



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA PODNIKATELSKÁ
ÚSTAV INFORMATIKY

FACULTY OF BUSINESS AND MANAGEMENT
INSTITUTE OF INFORMATICS

NÁVRH SQL DATABÁZE PRO PODPORU ČINNOSTÍ MALÉ IT FIRMY

PROPOSAL OF SQL DATABASE TO SUPPORT THE ACTIVITIES IN SMALL IT COMPANY

BAKALÁŘSKÁ PRÁCE

BACHELOR'S THESIS

AUTOR PRÁCE
AUTHOR

PETR ŘÍHA

VEDOUcí PRÁCE
SUPERVISOR

ING. JIŘÍ KŘÍŽ, PH.D.

BRNO 2012

ZADÁNÍ BAKALÁŘSKÉ PRÁCE

Říha Petr

Manažerská informatika (6209R021)

Ředitel ústavu Vám v souladu se zákonem č.111/1998 o vysokých školách, Studijním a zkušebním řádem VUT v Brně a Směrnicí děkana pro realizaci bakalářských a magisterských studijních programů zadává bakalářskou práci s názvem:

Návrh SQL databáze pro podporu činností malé IT firmy

v anglickém jazyce:

Proposal of SQL Database to Support the Activities in Small IT Company

Pokyny pro vypracování:

Úvod

Vymezení problému a cíle práce

Teoretická východiska práce

Analýza problému a současné situace

Vlastní návrhy řešení, přínos návrhů řešení

Závěr

Seznam použité literatury

Přílohy

Seznam odborné literatury:

KOCH, M.; NEUWIRTH, B. Datové a funkční modelování. 3. přepracované vydání. Brno: Akademické nakladatelství Cerm, 2008. 121s. ISBN 978-80-214-3731-9.

HOTEK, M. Microsoft SQL Server 2008: krok za krokem. 1.vydání. Brno: Computer Press, 2009. 488 s. ISBN 978-80-251-2466-6.

LACKO, L. Jak vyzrát na SQL Server 2008. Brno: Computer Press, 2009. 469 s. ISBN 978-80-251-2101.

MOLINARO, A. SQL: Kuchařka programátora. Brno: Computer Press, 2009. 576 s. ISBN 978-80-251-2617-2.

Vedoucí bakalářské práce: Ing. Jiří Kříž, Ph.D.

Termín odevzdání bakalářské práce je stanoven časovým plánem akademického roku 2011/2011.

L.S.

Ing. Jiří Kříž, Ph.D.
Ředitel ústavu

doc. RNDr. Anna Putnová, Ph.D., MBA
Děkan fakulty

V Brně, dne 08.05.2012

Abstrakt

Bakalářská práce se zaměřuje na návrh SQL databáze pro podporu činností malé IT firmy a její vytvoření v programu MS SQL Server.

Abstract

This thesis focuses on the design of SQL databases to support the activities of a small IT company and its creation in MS SQL Server.

Klíčová slova

Databáze, SQL, MS SQL Server, IT společnost

Key words

Database, SQL, MS SQL Server, IT company

Bibliografická citace práce

ŘÍHA, P. *Návrh SQL databáze pro podporu činností malé IT firmy*. Brno: Vysoké učení technické v Brně, Fakulta podnikatelská, 2012. 61 s., 10 s. příloh. Vedoucí bakalářské práce Ing. Jiří Kříž, Ph.D.

Čestné prohlášení:

Prohlašuji, že předložená bakalářská práce je původní a zpracoval jsem ji samostatně. Prohlašuji, že citace použitých pramenu je úplná, že jsem ve své práci neporušil autorská práva (ve smyslu Zákona č. 121/2000 Sb., o právu autorském a o právech souvisejících s právem autorským).

V Brně dne

.....

Petr Říha

OBSAH

Úvod	9
Vymezení problému a cíle práce.....	10
1 Teoretická východiska práce	11
1.1 Informace	11
1.2 Informační technologie	12
1.3 Databáze.....	13
1.3.1 Historie databází	13
1.3.2 Základní databázové pojmy	13
1.3.3 Základní pojmy relační databáze	15
1.3.4 Integrita relačního modelu	15
1.3.5 Metodika návrhu databáze	17
1.4 Jazyk SQL	19
1.4.1 Historie jazyka SQL.....	19
1.4.2 Základní datové typy jazyka SQL	19
1.4.3 Základní SQL příkazy.....	20
1.5 Seznámení s vývojem MS SQL server.....	21
2 Analýza současného stavu.....	22
2.1 Základní informace o společnosti	22
2.2 Historie společnosti.....	22
2.3 Organizační struktura	22
2.4 Oblast činnosti.....	23
2.5 Konkurence	23
2.6 Zákazníci	24
2.7 Dodavatelé a obchodní partneři	24
2.8 Swot analýza	24

2.9	Informační technologie firmy	25
2.9.1	Hardware a počítačová síť	25
2.9.2	Software	25
2.9.3	Zpracování, záloha a archivace dat	26
2.10	Informační systém	27
2.10.1	Analýza současného stavu a definice nedostatků	27
2.10.2	Legislativní požadavek na informační systém (databázi)	28
3	Vlastní návrh řešení, přínos návrhu řešení	29
3.1	Požadavky na nový systém	29
3.2	Analýza činností ve firmě	30
3.2.1	Vytvoření prodejní objednávky	30
3.2.2	Vytvoření nákupní objednávky	32
3.2.3	Přidání nového zaměstnance	32
3.2.4	Průběh servisem	34
3.3	DFD diagram	36
3.4	Konceptuální návrh databáze	37
3.4.1	Definice základních entit	37
3.4.2	Identifikace relací mezi základními entitami	38
3.4.3	Základní E-R diagram	39
3.5	Logický návrh databáze	40
3.5.1	Dekompozice entit obecně	40
3.5.2	Zákazník - zboží	41
3.5.3	Zaměstnanec - zboží	41
3.5.4	Pobočka - zboží	42
3.5.5	Schéma konečné dekompozice	43
3.5.6	Popis atributů entit	44

3.5.7	Finální E-R Diagram.....	50
3.6	Fyzický návrh databáze.....	51
3.6.1	Dotazy na výstup z vlastního řešení	51
3.7	Přínos návrhu řešení	56
Závěr		57
Literatura		58
	Knižní zdroje.....	58
	Online zdroje.....	58
Seznam použitých zkratk		59
Seznam obrázků		60
Seznam tabulek		61
Přílohy		62

Úvod

V dnešní době, kdy jsou informace a znalosti jedním z nejdůležitějších faktorů podnikání, je velice důležité nějakým způsobem tyto informace uchovávat. Efektivní uchovávání a zpracování informací se stává podstatným nástrojem k dosažení zisku.

Velice důležitým poznatkem je, že sice uchováváme informace, ale v systémech nebo prostředích, která jsou zmatená, nepřehledná či nejednotná a stávají se velkou překážkou v práci s informacemi. Tato bakalářská práce je tvořena za účelem efektivního, přehledného a bezpečného uchovávání dat a informací z malé IT. V této práci se tedy zaměřím na vývoj databáze pro podporu činností v malé IT firmě za užitím programovacího databázového jazyka SQL.

Databáze bude vyvíjena na platformě Microsoft SQL Server 2008 R2, která byla nejnovějším vývojovým nástrojem ve finální verzi od společnosti Microsoft v době zahájení prací na tomto projektu. V průběhu vytváření této práce společnost Microsoft uvedla verzi MS SQL Server 2012. Nicméně tyto verze jsou vzájemně kompatibilní a na funkčnost nebo návrh databáze to nebude mít žádný negativní dopad v případě přechodu na vyšší verzi, tedy verzi 2012.

Vymezení problému a cíle práce

Vymezení problému

Problémem, který budu v této práci řešit, je absence komplexního informačního systému, který by pomohl usnadnit vykonávání každodenních činností ve firmě SWODN STORE (společnost zabývající se obchodem, sestavováním PC sestav, vytvářením malých podnikových sítí a poradenstvím v oblasti IT).

Cíle práce

Cílem této práce je navrhnout a vytvořit SQL databázi pro podporu činností v této IT firmě. Dále následuje popsání některých procesů pomocí vývojových diagramů, vytvoření návrhu databáze a nakonec její fyzické vytvoření v prostředí MS SQL server 2008 R2.

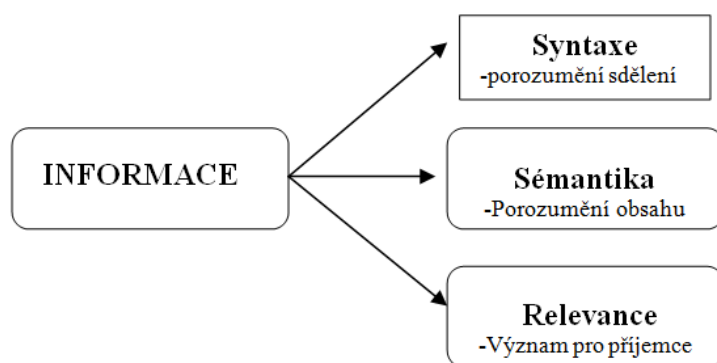
1 Teoretická východiska práce

Tento oddíl je zaměřen na teoretické podklady nutné k zvládnutí této práce a získání informací k porozumění dané problematiky.

1.1 Informace

V dnešní době jsou informace a vědomosti tím nejcennějším, co každá společnost může vlastnit. Jen tyto dvě položky jsou hlavním producentem úspěchu a zisku.

Na informaci můžeme nahlížet jako na zprávu či vjem, který splňuje tři základní požadavky. Prvním z nich je syntaktická relevance. Zde jde hlavně o to, aby subjekt, který zprávu přijímá, byl schopen ji správně detekovat a porozumět jejímu obsahu. Druhým požadavkem je sémantická relevance. Subjekt musí vědět co mu zpráva podává za informaci o něm a jeho okolním prostředí. Třetím a posledním požadavkem je pragmatická relevance. Zpráva musí mít pro subjekt nějaký význam [4].



Obr. 1: Tři základní požadavky na informaci

Zpracováno dle: Koch, Neuwirth, 2008, s. 4

Dále máme mnoho hledisek, dle kterých můžeme informace třídit. Například zda jsou informace krátkodobé, dlouhodobé, prognostické, aktuální, historické, operativní, strategické, taktické, ... [4, str.4]

1.2 Informační technologie

Abychom mohli zpracovávat data, ze kterých posléze vzniknou informace, potřebujeme k tomu určité nástroje, metody a znalosti, které se nazývají informační technologie. Tyto nástroje a metody sloužící ke zpracování dat se jinak nazývají hardware a software.

Hardware

Hardwarem se rozumí fyzické prvky technického vybavení počítače.

Hardware rozdělujeme do těchto oblastí [2]:

- počítače (osobní, servery, mainframe)
 - komponenty (grafická karta, procesor...)
- periferní zařízení (tiskárny, scanner apod.)
- komunikační technologie (sít'ové prvky, modemy apod.)

Software

Software nebo též programové vybavení počítače je sada všech programů umístěných v počítači. Software vzniká jeho naprogramováním a dá se rozdělit podle jeho funkce na [8]:

- *Systémový software*, který je programovým vybavením počítače umožňující spouštění nebo zpracování aplikačního software. Typickým představitelem systémového software je operační systém.
- *Aplikační software* je programové vybavení navržené a vytvořené pro řešení nějakého konkrétního problému nebo za konkrétním účelem.

1.3 Databáze

1.3.1 Historie databází

Je tomu už dlouho co naši předci nabyli potřeby ukládat informace k jejich dalšímu využití. Jak je známo informace je jedním z nejdůležitějších zdrojů, jak v podnikání, tak i v osobním životě. Dříve byly k ukládání dat využívány různé papírové kartotéky, které sloužily jako datové úložiště většinou pro pracovní účely, ale i pro archivaci dat. Například vzpomeňme na kartotéku u zubního lékaře, kde měl každý své zdravotní záznamy v papírové podobě (papírové složce). V těchto kartotékách byly data uloženy většinou podle abecedy. Další takovou papírovou kartotékou jsou knihovny. I knihovnu totiž lze chápat jako úložiště dat – v tomto případě se jedná o data v knihách, které jsou většinou stejně jako u lékaře seřazeny podle abecedy. Po té přišli na „scénu“ počítače a s nimi první myšlenky jak data zpřístupnit více lidem na více místech zároveň. Všechny tyto problémy se postupem času začalo dařit řešit s experimentováním s počítačovými programovacími jazyky. Nicméně žádný z klasických programovacích jazyků nebyl pro tvorbu databází zcela vhodný. Proto přišli první myšlenky na vytvoření „databázového jazyka“. V roce 1959, tak vznikl první databázový jazyk nazývaný COBOL (COmmon Business Oriented Language).

Dalším databázovým jazykem, který stavěl na základech, které položil COBOL byl jazyk Sequel, nyní SQL. *„S jazykem SQL (Structured Query Language) bylo možné se poprvé setkat již v roce 1974. Jeho první označení však nebylo SQL, nýbrž Sequel. Poprvé byl použit v Systému R vyvinutého v kalifornské laboratoři IBM. Postupem času vznikaly další „verze“ jazyka a byla potřeba jeho standardizace [10].“*

1.3.2 Základní databázové pojmy

Jedná se o pojmy, které používáme v databázovém světě, které odpovídají pojmům vyskytujícím se ve světě reálném.

Entita

„Je jedním z nejzákladnějších a nejdůležitějších pojmů. Entita v datovém modelu představuje datový objekt, který reprezentuje reálný datový objekt. Pod pojmem

reálný datový objekt si můžeme představit jakýkoliv prvek reálného světa (např. auto, kolo, počítač, člověk atd.), který lze popsat určitými charakteristikami (vlastnostmi).

Atribut

Atributem nazýváme v databázové terminologii charakteristiky, kterými popisujeme reálný datový objekt (např. název, výrobce, hmotnost, rok výroby atd.).

Vazba mezi entitami

Vazba mezi entitami je další pojem, který je potřeba představit. Jelikož každá entita odpovídá určitým prvkům z reálného prostředí, a tyto entity mezi sebou mají určitý vztah, rozeznáváme několik druhů vazeb. Tyto druhy vazeb uvedu na příkladech.

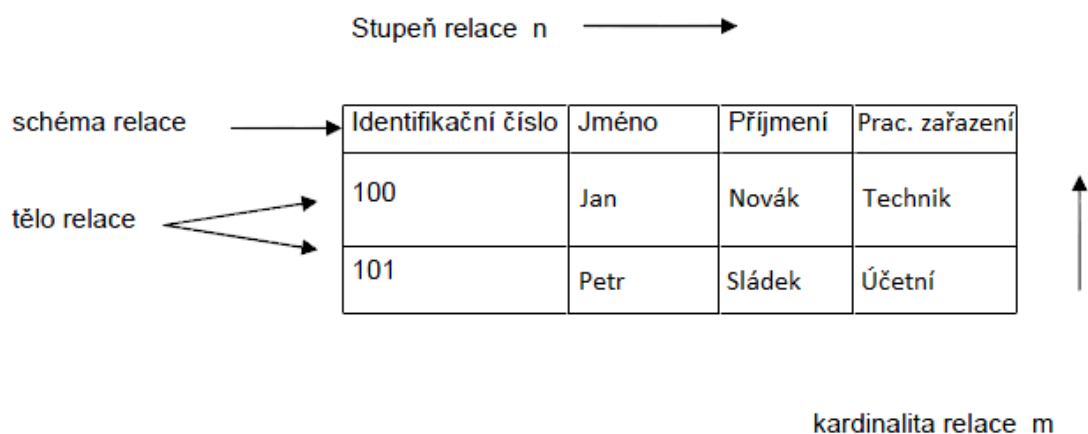
- **Vazba 1:1** - Zde lze jako vhodný příklad zvolit, že každý člověk má právě jedno rodné číslo a naopak zase každé rodné číslo náleží právě jednomu člověku.
- **Vazba 1:N** - Konkrétní člověk může vlastnit několik automobilů, nicméně každý konkrétní automobil může být vlastněn pouze jedním konkrétním člověkem.
- **Vazba N:1** - Je obdobou vazby 1:N. Příkladem může být, když několik lidí vlastní jeden dům.
- **Vazba M:N** - Posledním typem vazby je M:N. Tuto vazbu lze dostatečně charakterizovat na příkladu, kdy člověk ve firmě může zastávat více pozic a naopak, každou z těchto pozic může zastávat více osob.

Datový model

Datový model je pojem, který charakterizuje jakým způsobem jsou data reprezentována v databázi nebo informačním systému. Zpočátku byly používány dva typy modelů hierarchický a síťový. Hierarchický model je založen na modelování hierarchie mezi entitami pomocí vztahu nadřazenosti a podřízenosti, síťový, vychází z teorie grafu, kde uzly v grafu odpovídají entitám a orientované hrany definují vztahy mezi entitami. Po té co se v 70. letech objevily problémy s realizací a implementací vazby M:N v těchto modelech, se ukázalo, že jsou nedostatečné, a tak vznikl relační model, který se stal standardem a při tvorbě databází se používá dodnes [11]."

1.3.3 Základní pojmy relační databáze

Základním pojmem relačních databází je relace (entita). Popis relace (entity) je znázorněn na obrázku.



Obr. 2: Popis relace (entity)

Zdroj: vlastní zpracování dle: Koch, Neuwirth, 2008, s. 27

Schéma relace zapisujeme ve tvaru $R(A_1, A_2, \dots, A_n)$. Počet atributů n relace se označuje stupeň (řád) relace, kardinalita těla relace m se označuje kardinalita relace. A platí, že stupeň relace je konstantní, kardinalita proměnná.

Pravidla reprezentace relace [4, str.27]:

- „každý řádek odpovídá jedné n -tici relace
- význam každého sloupce je určen jménem atributu
- pořadí n -tic je nevýznamné
- pořadí sloupců je nevýznamné
- tabulka neobsahuje duplicitní n -tice
- tabulka neobsahuje duplicitní atributy
- hodnoty ve sloupcích jsou atomické“

1.3.4 Integrita relačního modelu

Dalším z důležitých pojmů vyskytující se v okruhu relačních databází je integrita. Integrita je stav, při kterém data uložená v modelu odpovídají vlastnostem

objektu reálného světa. Tyto integritní omezení dělíme na: integritní omezení pro entity a integritní omezení pro vztahy entit.

Integritní omezení pro entity [4, str. 28-29]:

- *„Kandidátní klíč je množina atributu relace, která má tyto vlastnosti:*
 - 1. Je jednoznačná, v dané relaci nejsou žádné dvě n-tice, které mají stejné hodnoty.*
 - 2. Je minimální, nelze vypustit žádný atribut, aniž by se porušilo pravidlo 1.*
- *Primární klíč (Primary key) – jeden z kandidátních klíčů se stává primárním klíčem, ostatní kandidátní klíče se stávají alternativními klíči.*
- *Cizí klíč (Foreign key) - je atribut, který má tyto vlastnosti:*
 - 1. Každá hodnota je buď plně zadaná, nebo plně nezadaná.*
 - 2. Existuje jiná relace s takovým primárním klíčem, že každá hodnota cizího klíče = hodnotě primárního klíče nějaké n-tice této jiné relace.“*

Integritní omezení pro vztahy entit:

Dalším nutným krokem je dekomponovat relace do vhodnějšího tvaru, tento krok se nazývá normalizace.

Při normalizaci jsou nutné dodržet tato pravidla:

- zachování bezztrátovosti při zpětném spojení
- zachování závislostí
- odstranění redundance

Existuje několik forem normalizace. Datový model je pak optimálně navržen pokud je splňuje. Nejdůležitější normalizační normy jsou [4]:

- **„První normální forma** - Relace je v první normální formě, neobsahuje-li složené či vícehodnotové atributy.
- **Druhá normální forma** - Relace je ve druhé normální formě, pokud je v první normální formě a všechny atributy jsou závislé na celém kandidátním klíči.
- **Třetí normální forma** - relace je ve třetí normální formě, pokud je ve druhé normální formě a navíc všechny její neklíčové atributy jsou vzájemně nezávislé.
- **Boyce – Coddova normální forma** - Relace je v Boyce-Coddove normální formě, pokud mezi kandidátními klíči není žádná funkční závislost a navíc splňují tyto podmínky:
 - relace má minimálně dva kandidátní klíče
 - nejméně dva z kandidátních klíčů jsou složené
 - kandidátní klíče se musí překrývat v některých attributech“

1.3.5 Metodika návrhu databáze

Jedná se o zmapování, jak postupovat při tvorbě efektivní databáze. Velmi podrobný popis postupu při návrhu databáze je uveden v knize Mistrovství - Databáze: Profesionální průvodce tvorbou efektivních databází.

„Konceptuální návrh databáze

KROK I. Vytvoření ER modelu

- *Identifikace entit*
- *Identifikace relací*
- *Identifikace a spojení atributu s entitami nebo relacemi*
- *Určení domén atributu*
- *Určení atributu, které budou kandidátními, primárními a alternativními klíči*
- *Specializace/generalizace entit (volitelně)*
- *Kontrola redundance v modelu*
- *Kontrola, zda model podporuje uživatelské transakce*
- *Posouzení konceptuálního návrhu databáze s uživateli*

Logický návrh databáze

KROK 2 Mapování ER modelu do tabulek

- *Vytvoření tabulek*
- *Kontrola tabulek pomocí normalizace*
- *Kontrola, zda tabulky podporují uživatelské transakce*
- *Kontrola integritních omezení*
- *Posouzení logického návrhu databáze s uživateli*

Fyzický návrh databáze

KROK 3 Převod logického návrhu databáze do cílového DBMS

- *Návrh podkladových tabulek*
- *Návrh reprezentace odvozených dat*
- *Návrh zbývajících integritních omezení*

KROK 4 Volba organizace souboru a indexu

- *Analýza transakcí*
- *Volba organizace souboru*
- *Volba indexu*

KROK 5 Návrh uživatelských pohledů

KROK 6 Návrh bezpečnostních mechanismů

KROK 7 Zvážení zavedení kontrolované redundance

KROK 8 Monitorování a doladění systému v provozu [1, str. 208]“

1.4 Jazyk SQL

SQL - Structured Query Language (Strukturovaný dotazovací jazyk) - je programovací dotazovací jazyk pro manipulaci, správu a organizování dat uložených v databázi situované ať na místním či vzdáleném počítači nebo také serveru [9].

1.4.1 Historie jazyka SQL

„S jazykem SQL (Structured Query Language) se můžeme setkat již v roce 1974. V té době se však jmenoval Sequel. Jako první byl použit v Systému R vyvinutého v kalifornské laboratoři IBM. Od té doby se jazyk rozšířil a byl používán v ostatních systémech. Postupem času však vznikaly další „verze“ jazyka a bylo potřeba, aby byl standardizován. K tomu došlo v roce 1986, kdy jej přijala standardizační skupina ANSI (a v roce 1987 ISO). Standardem byl uznán „dialekt“ firmy IBM. V literatuře se můžeme setkat také s označením SQL86. Později bylo potřeba rozšířit definičního jazyka pro možnost integritního omezení. Výsledná zpráva byla zveřejněna v roce 1989 organizací ISO. Tomuto rozšíření se říká SQL89. Poslední přijatý standard byl v roce 1992 (ANSI) a je označován jako SQL92 [10].“

1.4.2 Základní datové typy jazyka SQL

Při práci s databázemi, tedy při jejich vytváření, musí mít jednotlivé atributy v každé relaci určen datový typ, do kterého poté zadáváme data dle povahy a datového typu atributu. Tyto datové typy dělíme na [12]:

- řetězcové: CHARACTER(n), CHARACTER VARYING(n), VARYING(n)
- numerické
 - přesné - NUMERIC(p, q), DECIMAL(p, q),
 - přibližné - INTEGER, SMALLINT, FLOAT(p), REAL, DOUBLE PRECISION
- datum a čas: DATE, TIME, TIMESTAMP
- intervalové: INTERVAL
- booleovský: BIT, BOOLEAN

1.4.3 Základní SQL příkazy

Základní příkazy pro manipulaci s daty

DML (Data Manipulation Language) – příkazy, které umožňují získat data z databáze a databázi upravovat [12]:

- SELECT- slouží k výběru záznamu a k řazení dat, je to jeden z nejzákladnějších příkazů
- INSERT – slouží k vložení nových dat do databáze
- UPDATE – slouží ke změně dat v databázi
- DELETE – slouží k odstranění záznamu z databáze

Základní příkazy pro definici dat

DDL (Data Definition Language) - příkazy pro úpravu struktury databáze. Tím se myslí úpravu tabulek, indexů nebo pohledů [12]:

- CREATE – slouží k vytváření nových objektů
- ALTER – slouží ke změně či úpravě existujících objektů
- DROP – slouží k odstraňování objektů

1.5 Seznámení s vývojem MS SQL server

„První historické zmínky o SQL Serveru se začaly psát v roce 1988, tehdy tento produkt ještě dodávala společnost Sybase. Tento produkt byl vyvíjen pro operační systém OS/2. V roce 1993 nastal průlom, kdy firma Sybase uvedla verzi SQL Serveru 4.2, což byla klasická desktopová databáze pro kanceláře a malé firmy určená pro operační systém Windows. V roce 1994 koupil tento produkt Microsoft a začal ho vyvíjet podle svého. První verzí pocházející z vývojových center Microsoftu byl v roce 1995 SQL server 6.05 primárně určený jako databázový produkt do segmentu small business. Vzrostl výkon a tuto verzi bylo možné využívat i pro internetové aplikace. Verze SQL Server 6.5, která byla uvedena v roce 1996 byla určena pro platformu Windows. Verzi SQL Server 7.0., která přišla na trh v roce 1998, bylo možné označit přívlastkem „webová databáze“. U této verze bylo kompletně přepsané a optimalizované jádro. Tento produkt byl první, který konečně začal konkurovat databázím Oracle a IBM DB2, kde na jeho straně stála velice příznivá cena. Ve verzi SQL Server 2000 se poprvé setkáváme s podporou Business Intelligence. SQL Server 2005 představoval významnou inovaci v oblasti Business Intelligence, a hlavně v užití XML jako nativního datového typu [5, str. 15].“

V současné době je na trhu verze SQL server 2008 R2. S touto verzí bude pracováno při vývoji databáze tvořené v této práci. Nicméně ještě před dokončením společnost Microsoft uvede nejnovější verzi a to SQL server 2012, která přináší další možnosti a rozšíření.

2 Analýza současného stavu

2.1 Základní informace o společnosti



Obr. 3: Logo společnosti

Zdroj: firemní dokumenty

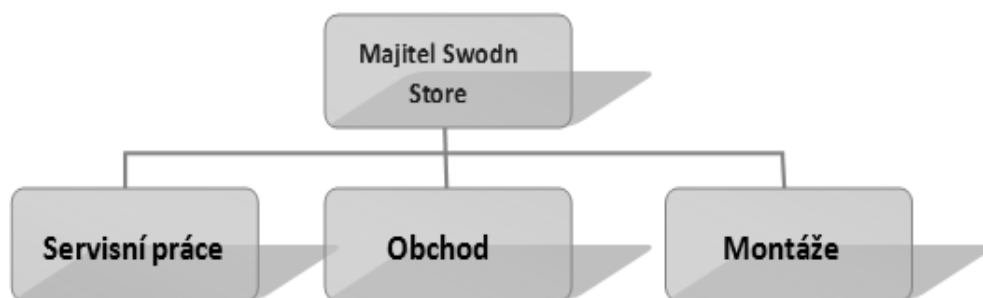
Název společnosti:	SWODN STORE
Právní forma:	fyzická osoba
IČO:	87561531
Sídlo:	náměstí ČSA 56, 67602 Moravské Budějovice
Den zápisu do OR:	25. ledna. 2011
Web:	www.swodn.cz

2.2 Historie společnosti

Společnost byla založena 25. ledna 2011 zapsáním do živnostenského rejstříku za účelem podnikání Jiřím Svobodou. Společnost Swodn započala své působení 1. února 2011. Nyní má společnost 3 zaměstnance včetně samotného majitele pana Svobody. Pan Svoboda má již mnohaletou praxi v oboru IT a obchodu, kde již působil v konkurenční společnosti zhruba po dobu 9 let. Mezi významné partnery společnosti patří 100megadistribution s.r.o., ESET spol. s.r.o., EDsystem a Setos telefony.

2.3 Organizační struktura

Společnost je vedená panem Svobodou, který má pod sebou dva zaměstnance. Sám pan Svoboda se dá nazvat vedoucím obchodu a styku se zákazníky. Další dva zaměstnanci se starají o servisní práce a montáže. Ve zbylém volném čase pracují na vývoji webových stránek.



Obr. 4: Organizační struktura

Zdroj: vlastní zpracování

2.4 Oblast činnosti

Hlavní sortiment služeb poskytovaný podnikem SWODN STORE:

- Zprostředkování obchodu a služeb v oblasti IT
- Poskytování software
- Poradenství v oblasti IT
- Servis mobilních telefonů a počítačů
- Projektování a výstavba menších počítačových sítí
- Sestavování PC na míru

2.5 Konkurence

V oblasti podnikání, která se zaměřuje na výpočetní techniku, můžeme v dnešní době nalézt spoustu podnikatelů, neboť tato doba zažívá rozmach IT. V lokalitě, kde podnik působí, se nachází několik různých konkurentů, avšak aktuální situace a nabídka těchto společností je neustále sledována s cílem, aby byla dosažena výhodná nabídka pro koncové zákazníky.

2.6 Zákazníci

Snahou podniku je působit na co největší oblast koncových zákazníků. Služby v oblasti výpočetní techniky v dnešní době využívá každá skupina obyvatelstva. Z prodeje můžeme usoudit, že mezi zákazníky z řad fyzických osob se nejčastěji vyskytují lidé mezi 25 a 40 lety, nejčastějšími zákazníky z oblasti právnických osob jsou malé firmy. Firma Swodn rovněž obstarává vybavení škol.

2.7 Dodavatelé a obchodní partneři

Dodavatele podniku tvoří několik společností. Hlavním dodavatelem a také významným partnerem je společnost 100megadistribution s.r.o. Dalšími dodavateli jsou společnosti eD' system Czech, a.s, Eset, Asus, HP, LG, Dell, Samsung, Epson, Nokia, HTC a další.

2.8 Swot analýza

STRENGTHS (Silné stránky)

- Vysoká kvalifikace pracovníků
- Silné postavení na trhu (lokálním trhu v místě působnosti společnosti)
- Dokonalá znalost konkurence v okolí
- Kvalita odvedené práce
- Dobré jméno pana majitele z dob působení u konkurenční společnosti => velké množství stálých zákazníků

WEAKNESS (Slabé stránky)

- Slabá propagace firmy v okolí
- Absence e-shopu
- Málo zaměstnanců

OPORTUNITY (Příležitosti)

- Expandovat do širšího okolí
- Více propagovat firmu v místní TV a místním tisku
- Vybudování e-shopu

THREATS (Hrozby)

- Sezónní odliv zákazníků (letní dovolená)
- Ekonomická krize (stagnace vývoje)
- Rostoucí trend nákupu substitutů

2.9 Informační technologie firmy

Společnost Swodn řeší správu informačních technologií z vlastních zdrojů, tj. některý ze zaměstnanců vždy dostane na starost konkrétní problém, který je potřeba vyřešit. Hlavní úkoly nad správou a zabezpečením informačního systému si ovšem prozatím obstarává majitel sám.

2.9.1 Hardware a počítačová síť

Vezmeme-li v úvahu velikost firmy, lehce si uděláme představu o stavu hardware a počítačové sítě LAN. Jelikož se firma zabývá sestavováním PC na míru, tak všechny pracovní stanice pocházejí z vlastní činnosti. U těchto sestav jsou všechny komponenty vybírány s ohledem na optimální výkon pro všechny činnosti prováděné firmou Swodn a zároveň odpovídající cenu takového řešení, které zajišťuje, že stanice dodávané externí firmou by byly krajně nevýhodné řešení. Místní počítačová síť je řešena přes router pro pracovní stanice a jeden wifi router pro připojení pracovních notebooků. Dodavatelem internetového připojení je místní kabelová televize, která nabízí výhodné podmínky jak pro firmy, tak i pro domácnosti.

2.9.2 Software

Softwarová vybavenost ve firmě je na dobré úrovni a kvůli rychlému vývoji IT technologií zde najdeme opravdu pestrou škálu všeho softwaru. Vzhledem k tomu, že všechny stanice byly sestavovány se začátkem působení firmy Swodn na trhu, jsou všechny stanice moderní a zvládají všechny nejmodernější software. Všechny software, který firma Swodn využívá je pořízený legálně.

Všechny pracovní stanice, včetně notebooků, běží na platformě *Microsoft Windows 7 Profesional*. Jedna pracovní stanice využívá dual boot, kde jako druhý operační systém disponuje *Microsoft Windows XP Profesional*, kvůli zpětné kompatibilitě se staršími systémy. Na jedné sestavě jsou *Windows XP* instalovány primárně.

Neodmyslitelnou součástí softwarového vybavení je i kancelářský balík *Microsoft Office 2007*, který společnost zakoupila ve verzi Enterprise. Společnost pro svůj provoz využívá systému *S3 money* a pro přehled o nabízených produktech aktuálních cenách a stavu skladových zásob komplexního e-shop systému od společnosti *100mega distribution s.r.o.*

2.9.3 Zpracování, záloha a archivace dat

Neodmyslitelnou součástí práce s výpočetní technikou je záloha dat. Firma pracuje se dvěma typy dat a to data z vlastní činnosti důležitá pro fungování společnosti a pak data zákazníků, která se zálohují z důvodu prevence ztráty dat při servisních pracích. Vlastní data společnosti se archivují vždy jednou týdně v sobotu v poledne přes automatický systém. Majitel pak každé pondělní ráno kontroluje správnost a úplnost zálohovaných dat. K archivaci dat se majitel společnosti rozhodl využít interval 6 měsíců. Archivace se provádí dvojmo na externí média, jedna část uložena v bezpečí ve firmě a druhá mimo firmu z důvodu prevence ztráty dat živelnou pohromou nebo z jiných příčin.

2.10 Informační systém

Velice důležitou roli hrají v každé firmě informace. Stejně jako všude jinde, tak i zde se s informacemi nakládá jako s tím nejcennějším co firma má a může získat a pomocí různých interních směrnic a pravidel se zajišťuje jejich efektivní zpracování. K efektivnímu zpracování, uchovávání a zálohování nestačí jen vytvoření pravidel a směrnic, ale nezbytnou roli zde hraje i samotný systém.

2.10.1 Analýza současného stavu a definice nedostatků

Hlavním problémem ve firmě *Swodn* je nejednotnost a neefektivnost informačního (databázového) systému. Většina firemních dokumentů a informací je vytvořena v aplikace MS Office Word nebo Excel a uchovávána v odpovídajícím formátu a další část, ta s informacemi o stavu skladu a zaměstnancích, je evidována systémem S3 money. Zde bych navrhoval sjednocení do jednoho informačního systému nebo alespoň dostatečně zefektivnit tento systém, který je již ve společnosti pořízen. Ve výsledku bude asi nejlepší řešení ponechat stávající personálně/mzdový modul s3 money a provázat jej s vlastním návrhem řešení.

Nedostatky stávajícího systému:

- nejednotnost
- nízká efektivita
- nepřehlednost
- nebezpečí ztráty dat

Výhody pořízení serveru s SQL databází:

- zvýšení efektivity stávajícího systému možností propojení se systémem S3 money
- sjednocení všech údajů pod jeden komplexně propojený systém
- téměř nulové náklady na zlepšení stávajícího stavu s využitím již pořízeného systému (nutno pořídit server na platformě MS server 2008, s ohledem na předmět působení společnosti se jedná o relativně nízkou částku).

2.10.2 Legislativní požadavek na informační systém (databázi)

Podstatou všech databází je uchovávat informace. Ani databáze tvořená v této práci nebude výjimkou a je tedy nutno vědět jaké údaje o zákaznících, zaměstnancích, dodavatelích atd. můžeme, a jaké naopak nemůžeme uchovávat.

Zákon č. 101/2000 Sb., o ochraně osobních údajů

Tento zákon pojednává o ochraně osobních údajů, která je dána Listinou základních práv a svobod (právo na ochranu občana před neoprávněným zásahem do soukromí, neoprávněným shromažďováním, zveřejňováním nebo jiným zneužíváním osobních údajů).

3 Vlastní návrh řešení, přínos návrhu řešení

3.1 Požadavky na nový systém

Podstatou návrhu vlastního řešení je vytvoření databázové struktury nového systému. Od správnosti, efektivity a důkladnosti zpracování se dále budou odvíjet další kroky, navazující na tuto práci, k vytvoření efektivního a relativně jednoduchého informačního systému.

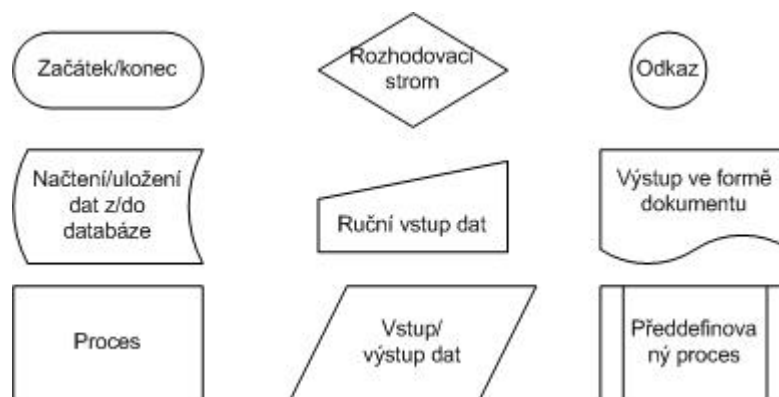
Struktura navrhovaného databázového systému je zaměřena na zpracování, evidenci, archivaci a údržbu uchovávaných dat. Tento výčet operací musí zahrnutou především činnosti jako jsou řízení objednávek, jak nákupních, tak prodejních. Evidence zaměstnanců a oprav respektive reklamací (servis). Tyto činnosti jsou popsány v podkapitole níže za pomoci vývojových diagramů.

Návrh databáze musí splňovat následující požadavky

- Možnost uchovávání důležitých provozních dat
- Přehlednost těchto dat
- Zabezpečení přístupu uživatelů jednotlivých oddělení a poboček jen k datům, ke kterým mají oprávnění a k žádným jiným
- Zajištění aktuálnosti dat
- Možnost propojení se stávajícím mzdovým a účetním modulem systému S3Money
- Jednoduchost

3.2 Analýza činností ve firmě

Pro potřebu návrhu struktury vytvářené databáze je nutné popsat stěžejní činnosti ve firmě. Tyto činnosti již byly zmíněny výše, ale pro přehlednost jsou to činnosti zahrnující přidání nového zaměstnance, vytvoření nákupních a prodejních objednávek průběh oprav respektive reklamací (servis).



Obr. 5: Použité značky vývojových diagramů

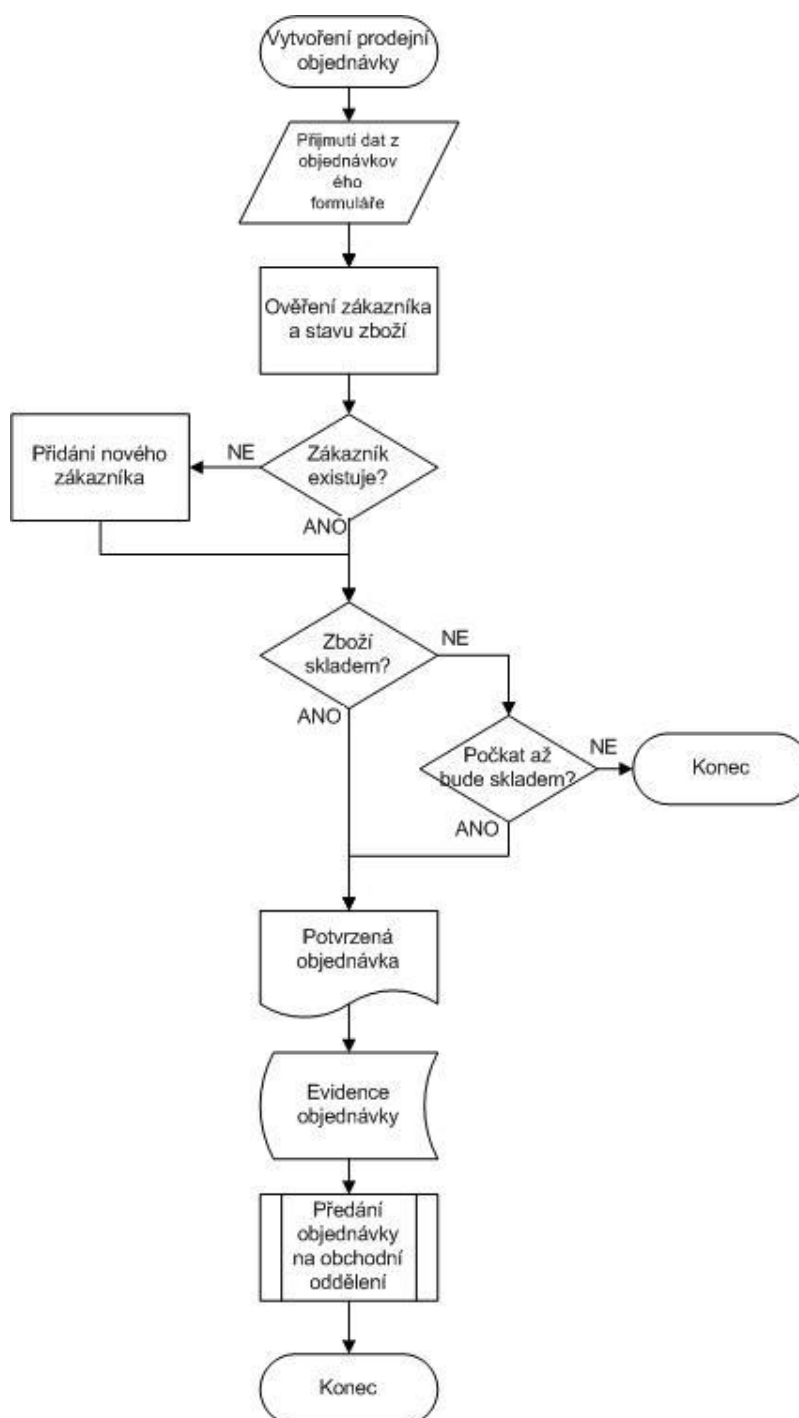
Zdroj: vlastní tvorba

3.2.1 Vytvoření prodejní objednávky

Jedná se o nejčastěji se vyskytující činnost ve firmě. Tato činnost zachycuje průběh, kdy zákazník poptává nějaké zboží. V současné době se objednávky dají realizovat osobní návštěvou, telefonicky nebo přes email. Do budoucna se počítá se zřízením e-shopu. Vytvářet objednávku telefonicky nebo po e-mailu je možno pouze registrovaným zákazníkům.

Prvním krokem po přijetí objednávky je ověření zákazníka a stavu zboží. Při ověření zákazníka se zjišťuje především zda se jedná již o registrovaného (stálého) zákazníka nebo nového, případně zda nemá nějaké neuhrazené objednávky. Tyto informace se jednoduše ověří z databázového systému. Dále se ověřuje zboží. Jde hlavně o množství, které je k dispozici na pobočce, případně pokud ho není dostatek, tak doba dodací lhůty a cenová nabídka. Dále je možnost pokud není zboží skladem objednávku zaregistrovat do systému a dodat ji v konkrétní dodací lhůtě.

Pokud je již takto vytvořená objednávka odsouhlasena zákazníkem, tak následuje její zaevidování do databáze a v případě, že ji vytvářelo jiné oddělení (jiný zaměstnanec) než obchodní, tak její postoupení obchodnímu oddělení, které se o realizaci objednávky postará.



Obr. 6: Vytvoření prodejní objednávky

Zdroj: vlastní tvorba

3.2.2 Vytvoření nákupní objednávky

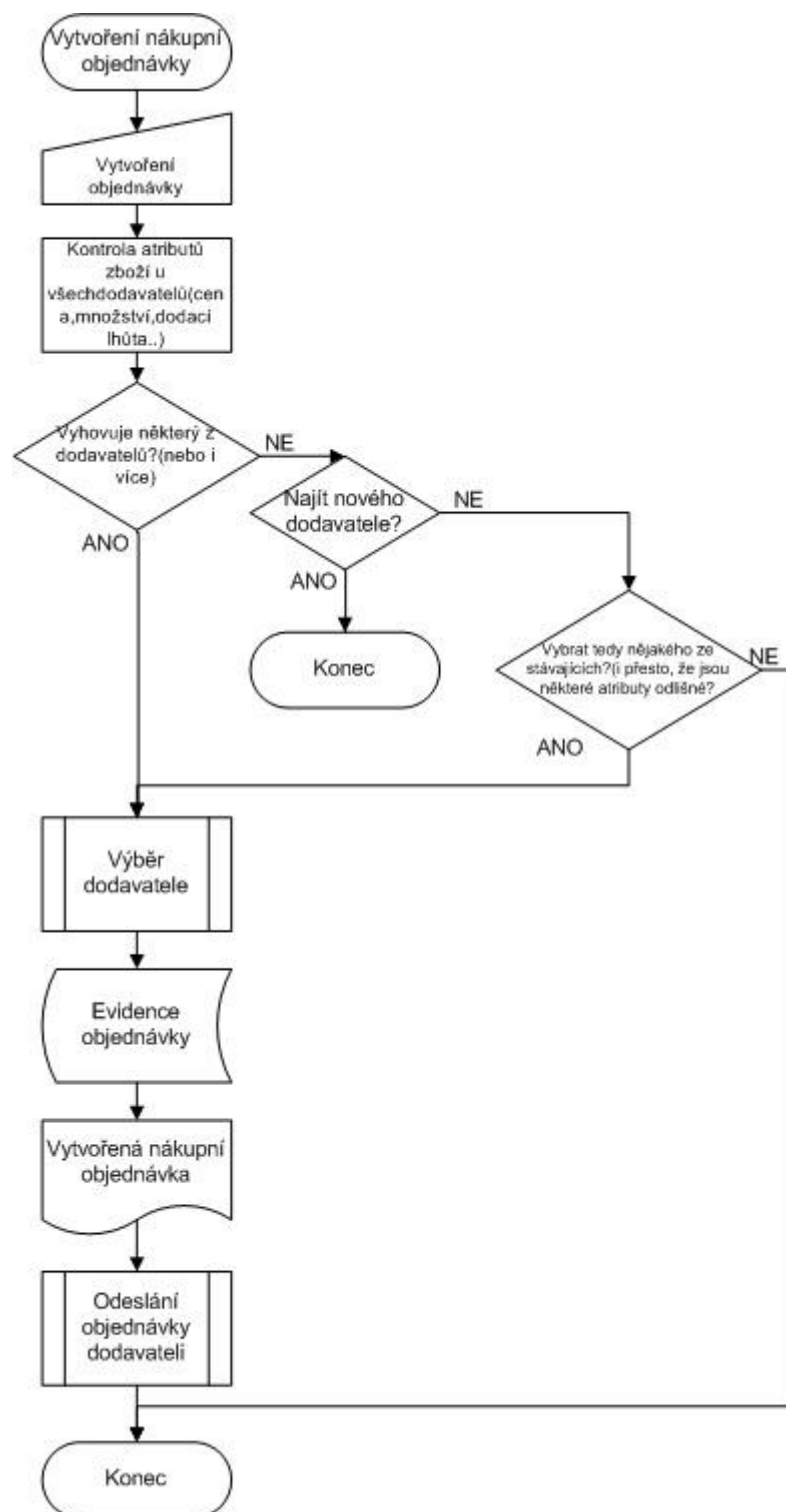
Tato činnost zachycuje vztah mezi firmou a dodavatelem, kdy firma poptává nějaké zboží. Všechny nákupní objednávky jsou určeny dlouhodobým dodavatelům. Následně se tedy ověří požadované atributy u požadovaného zboží, jako např.: cena, dodací lhůta, množství, které je schopen každý z dodavatelů poskytnout. V případě potřeby či nutnosti je možno najít i jiného dodavatele, s kterým pak může být navázána i dlouhodobější spolupráce, případně může být využit k pokrytí konkrétní dodávky (objednávky). Po té následuje konečná volba dodavatele, zaevidování objednávky do systému a její následné odeslání dodavateli.

V podmínkách s dodavatelskými společnostmi je dáno, že jakmile jim je objednávka vystavena a odeslána, je možno ji změnit pouze telefonicky a to nejméně 24 hodin před jejím dodáním.

3.2.3 Přidání nového zaměstnance

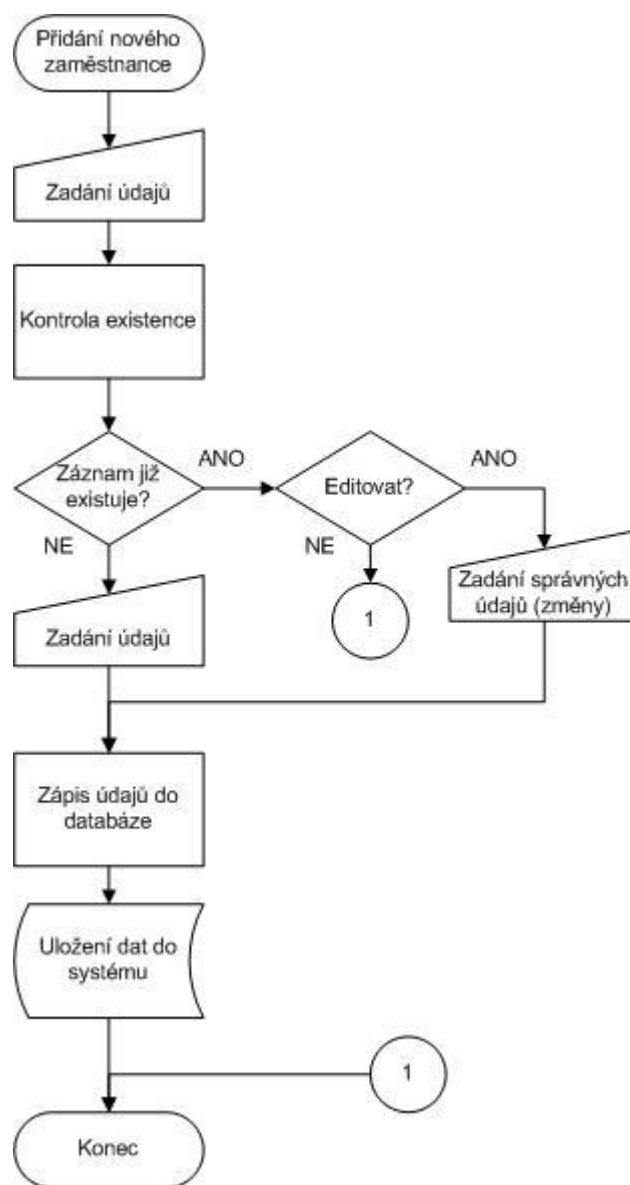
Jedná se o činnost, která je vzhledem k velikosti firmy prováděna málokdy, ale z hlediska správnosti vkládaných údajů s ohledem na propojení s mzdovým modulem je na ni kladena vysoká důležitost.

Po zadání klíčových atributů se ověří existence záznamu o tomto zaměstnanci ve firmě. Pokud již existuje je možnost jej editovat a nebo proces ukončit. V opačném případě následuje zadání všech potřebných údajů. Vizuální kontrola zadaných dat a následné uložení dat do systému.



Obr. 7: Vytvoření nákupní objednávky

Zdroj: vlastní tvorba



Obr. 8: Přidání nového zaměstnance

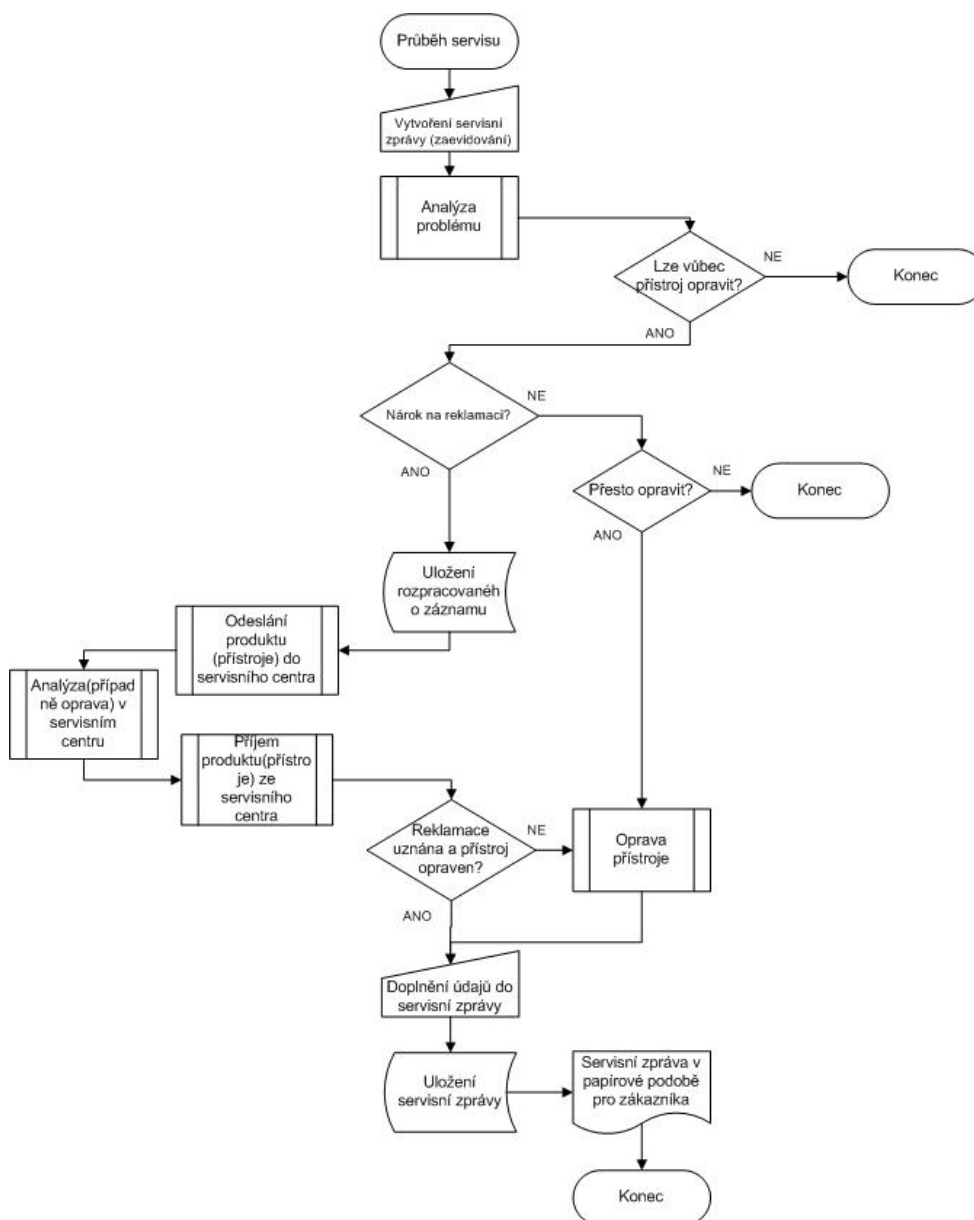
Zdroj: vlastní tvorba

3.2.4 Průběh servisem

Tato činnost zachycuje průběh oprav případně reklamací zboží (přístrojů, komponent). Zákazník sdělí servisnímu oddělení povahu problému. Pracovník oddělení problém zaeviduje jako servisní zprávu, provede analýzu přístroje. Zhodnotí zda je vůbec problém řešitelný a zjistí zda má zákazník nárok na reklamaci. Pokud ano je záznam uložen a produkt (přístroj) je odeslán konkrétnímu výrobcí nebo do jeho

autorizovaného servisního centra. Tam ještě jednou zanalyzují přístroj a reklamaci buď uznají nebo produkt vrátí s informací o neuznání reklamace. Pokud nárok na reklamaci není vůbec nebo nebyla výrobcem uznána, řeší se, zda chce zákazník přístroj opravit. Pokud ne, tak nastává konec procesu, pokud ano, je přístroj opraven, doplněna servisní zpráva kde je případně vytvořena i prodejní objednávka na náhradní díly nutné k opravě.

Po doplnění těchto údajů dojde k uložení servisní zprávy do databáze a následnému vytištění servisních záznamů pro zákazníka.



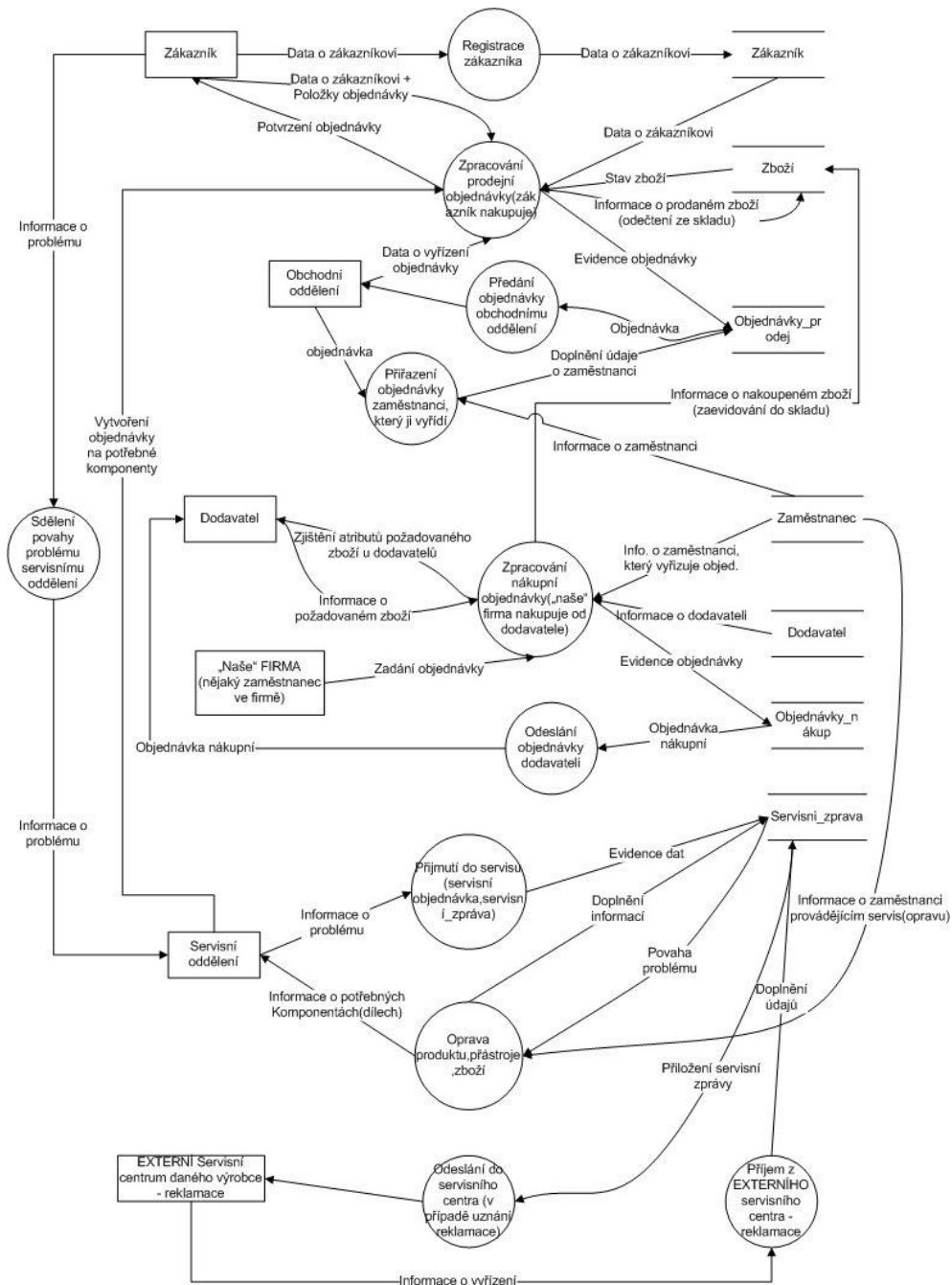
Obr. 9: Průběh servisu

Zdroj: vlastní tvorba

3.3 DFD diagram

Cílem DFD diagramu je popsat funkčnost navrhovaného systému. Funkčností se rozumí chování systému, které musí odrazet dění v reálném systému.

DFD diagram znázorněný níže popisuje vlastní návrh řešení informačního systému společnosti. Znázornění diagramu bylo použito dle notace Yourdan and Coad.



Obr. 10: DFD diagram celého systému

Zdroj: vlastní tvorba

3.4 Konceptuální návrh databáze

Konceptuální návrh databáze zahrnuje vytvoření (definování) základních entit a identifikaci relací (vztahů) mezi nimi. Po zpracování těchto dvou úkolů bude možno vytvořit základní E-R diagram.

3.4.1 Definice základních entit

Podstatou tohoto kroku je vytvoření základních entit (tabulek) databáze, se kterými se v praxi počítá. Tento počet tabulek ovšem není konečný. Počet tabulek bude daleko větší po provedení dekompozice a normalizace. Dále je nutno počítat ještě s několika číselníky.

Tabulka 1: definice základních entit

název entity	popis entity
dodavatelé	dodavatelé od kterých firma nakupuje své zboží (to co prodává)
nákupní objednávky	evidence objednávek kdy firma nakupuje od svých dodavatelů
pobočky	evidence poboček, které firma má/bude mít
pracovní pozice	pozice, které jednotlivý zaměstnanci zastávají
prodejní objednávky	evidence objednávek kdy firma prodává zákazníkům
servisní zpráva	evidence oprav a reklamací
zákazníci	zákazníci společnosti, kterým firma prodává nebo poskytuje služby
zaměstnanci	zaměstnanci pracující pro firmu
zboží	zboží, které firma prodává/nakupuje

Zdroj: vlastní tvorba

3.4.2 Identifikace relací mezi základními entitami

V této části se identifikují relace mezi základními entitami. Relace jsou identifikovány v tabulce na další stráně. Tabulka obsahuje název entit, které jsou ve vzájemném vztahu, typ relace mezi nimi a popis.

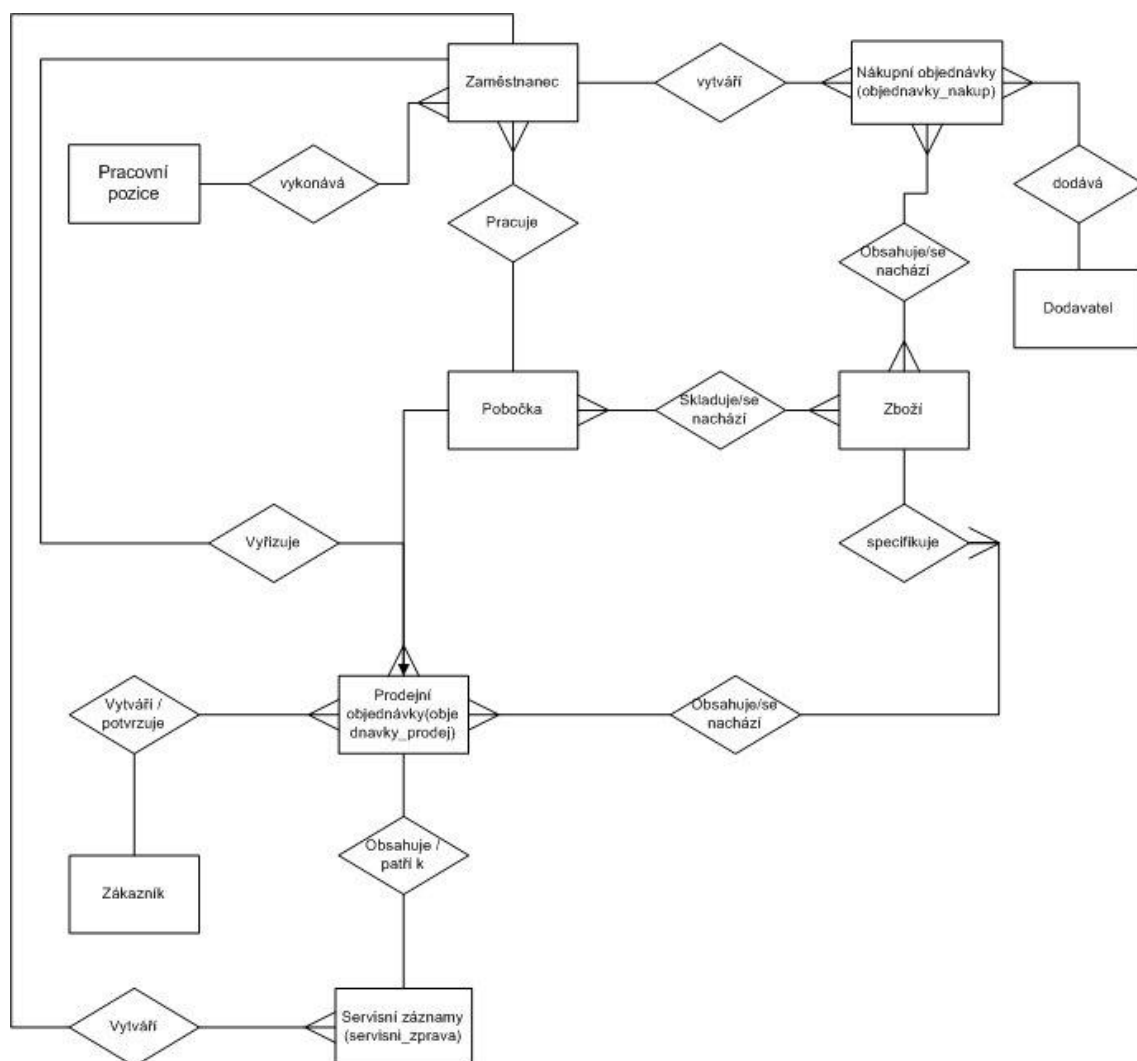
Tabulka 2: identifikace relací mezi základními entitami

Entity	typ relace	popis
prodejní objednávky - zákazník	N:1	zákazník může mít několik objednávek, ale konkrétní objednávka náleží jen 1 zákazníkovi
prodejní objednávky - zaměstnanec	N:1	zaměstnanec může vytvořit několik objednávek, jednu konkrétní objednávku tvoří 1 zaměstnanec
prodejní objednávky - zboží	M:N	na objednávce může být několik druhů zboží, konkrétní zboží se může vyskytovat na několika objednávkách
prodejní objednávky - servisní zpráva	1:1	k jedné servisní zprávě náleží max. 1 objednávka
zaměstnanec - pracovní pozice	1:N	zaměstnanec zastává jen jednu pozici, konkrétní pozici může vykonávat více zaměstnanců.
zaměstnanec - servisní zpráva	N:1	zaměstnanec tvoří několik servisních zpráv, konkrétní zprávu však tvoří pouze jeden
zaměstnanec - pobočka	1:N	zaměstnanec pracuje pro jednu pobočku, na jedné pobočce několik zaměstnanců
nákupní objednávky - dodavatel	1:N	dodavatel řeší několik objednávek, jedna konkrétní objednávka je však určena jen pro jednoho dodavatele
nákupní objednávky - zaměstnanec	1:N	zaměstnanec tvoří několik objednávek, konkrétní objednávku tvořil jen jeden
nákupní objednávky - zboží	M:N	na objednávce může být několik druhů zboží, konkrétní zboží se může vyskytovat na několika objednávkách
pobočka - zboží	M:N	na jedné pobočce je možno skladovat více druhů zboží, konkrétní zboží může být skladováno na více pobočkách současně

Zdroj: vlastní tvorba

3.4.3 Základní E-R diagram

V tomto kroku je definován E-R diagram tvořen pouze základními entitami (tabulkami). Nachází se zde mnoho relací M:N, které bude nutno rozložit a tím společně s úpravou tabulek do norem jejich počet vzroste.



Obr. 11: Základní E-R diagram

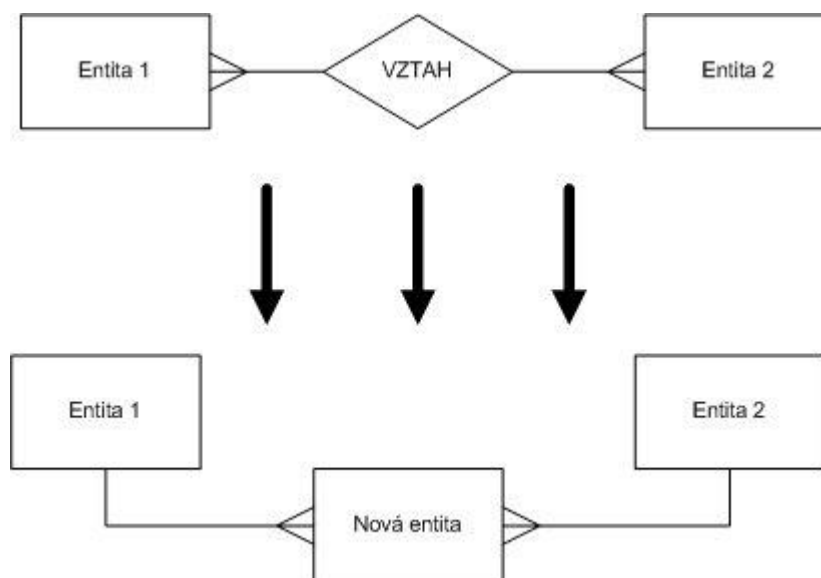
Zdroj: vlastní tvorba

3.5 Logický návrh databáze

Do této části návrhu řešení spadá dekompozice vazeb M:N, definice integritních omezení a atributy entit. Dále byly definovány datové typy jednotlivých atributů. Pro každou z těchto entit byla vytvořena tabulka, kde její název odpovídá názvu entity. Každá z těchto tabulek obsahuje integritní omezení, název položky (atributu), datový typ a rozsah, doplňující podmínky (default, not null, ...) a poznámky.

3.5.1 Dekompozice entit obecně

Obchodní oddělení jednotlivých poboček přijímají objednávky od zákazníků, vystavují objednávky pro své dodavatele (nakupují od nich) a evidují zásoby, které mají na svých skladech, tak při propojení těchto entit do požadovaných tvarů vznikají vazby N:M. Tyto vazby je nutno dekomponovat (rozložit) vytvořením pomocných tabulek.



Obr. 12: Obecné schéma dekompozice

Zdroj: vlastní tvorba

3.5.2 Zákazník - zboží

Zákazník objednává zboží od společnosti. Již dříve bylo zmíněno, že k realizaci objednávky telefonicky nebo přes e-mail je vyžadována registrace. Proto je tedy pro zákazníka výhodnější si vytvořit registraci i při běžném nákupu na prodejně, aby o tuto výhodu nebyl ochuzen.

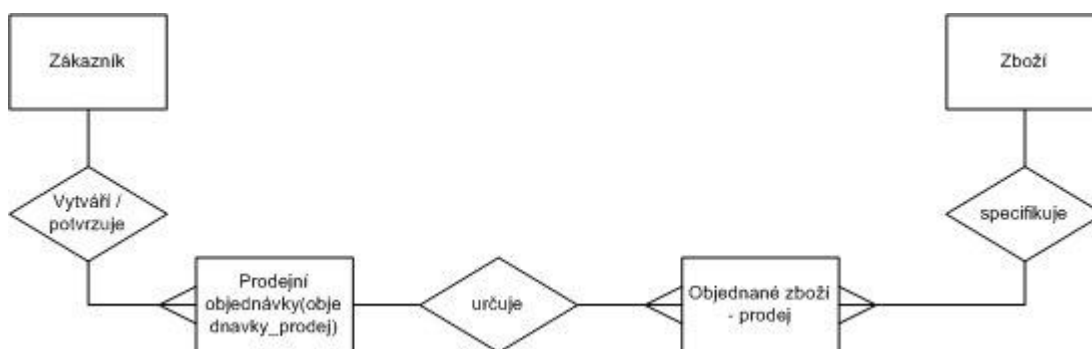
K samotnému vztahu entit zákazník a zboží, lze dodat, že každý zákazník si může objednat jakékoliv zboží a jeho varianty a naopak zase konkrétní druh zboží si může objednat více zákazníků. Jedná se tedy o vazbu M:N, kterou je nutno rozložit.



Obr. 13: Vztah entit zákazník - zboží

Zdroj: vlastní tvorba

Tyto objednávky se evidují do databáze jako objednané zboží - prodej. Zde právě vzniká vazba N:M, protože každý zákazník může mít N objednávek a zároveň zboží může být zaevidováno na více objednávkách ať už pro stejného zákazníka nebo pro nějakého jiného.



Obr. 14: Dekompozice entit zákazník - zboží

Zdroj: vlastní tvorba

3.5.3 Zaměstnanec - zboží

Zde se jedná o podobný problém jako ve vztahu zákazník - zboží. Zaměstnanec, Pro upřesnění zde popíši vztah zaměstnanec - pobočka. Nákupní objednávku vytváří

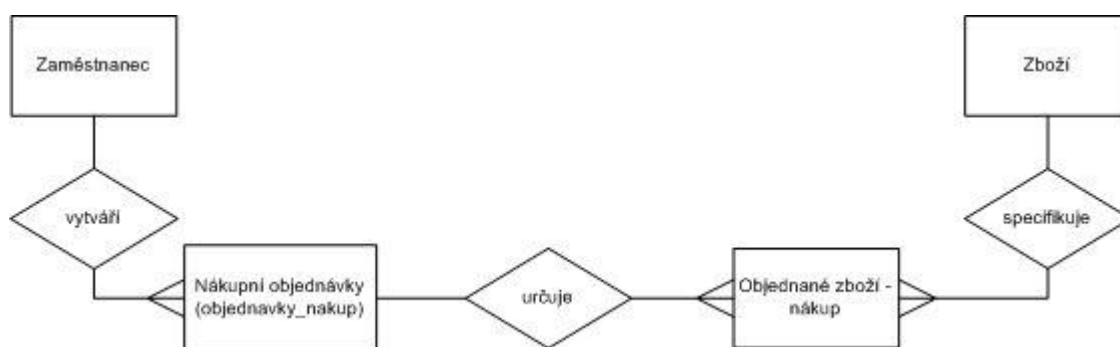
zaměstnanec, který náleží pouze jedné pobočce tudíž mezi zaměstnancem a pobočkou je vazba N:1. Zaměstnanec pracuje pouze na jedné pobočce a naopak na jedné konkrétní pobočce pracuje N zaměstnanců.

Zaměstnanec tedy objednává zboží od dodavatele, kde existuje mnoho variant zboží a může objednávat i více druhů tohoto zboží či nějaké jiné. Z opačné strany, z pohledu zboží, se zboží může vyskytovat na více nákupních objednávkách pro více poboček. Poboček proto, že zaměstnanec je pevně spjat s pobočkou.



Obr. 15: Vztah entit zaměstnanec - zboží

Zdroj: vlastní tvorba



Obr. 16: Dekompozice entit zaměstnanec - zboží

Zdroj: vlastní tvorba

3.5.4 Pobočka - zboží

Na různých pobočkách se uchovává stejný nebo podobný druh případně sortiment zboží. Zde je tedy také nutná dekompozice vazby M:N, pomocí entity zasoby_pobocky (zásoby na pobočkách) kde vzniknou vazby 1:N.



Obr. 17: Vztah entit pobočka - zboží

Zdroj: vlastní tvorba

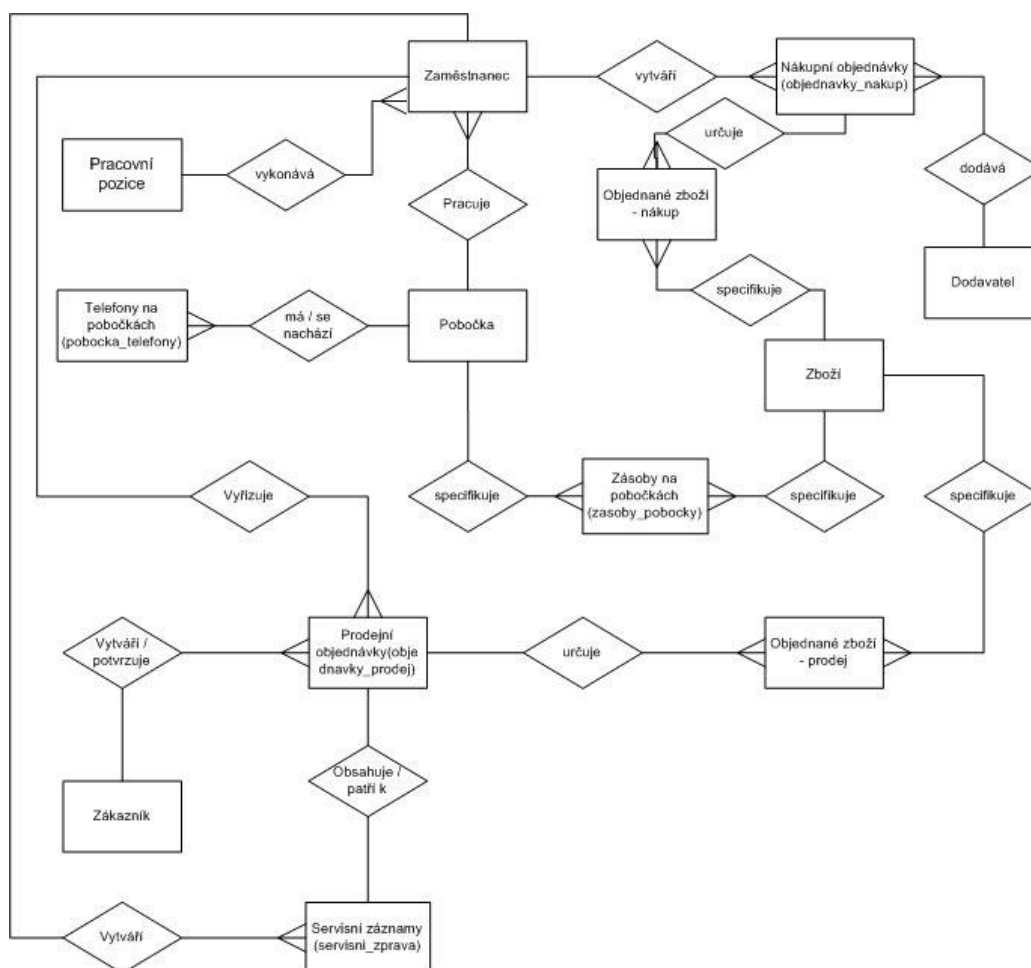


Obr. 18: Dekompozice entit pobočka - zboží

Zdroj: vlastní tvorba

3.5.5 Schéma konečné dekompozice

Na tomto schématu jsou k vidění všechny vazby vyskytující se v databázi, včetně těch dekomponovaných. Jedná se o finální shrnutí sloužící ke kompletnímu přehledu o vztazích mezi entitami.



Obr. 19: Schéma konečné dekompozice

Zdroj: vlastní tvorba

3.5.6 Popis atributů entit

Tento oddíl poskytuje přehled o názvech jednotlivých atributů entit, datových typech a rozsahů, případné doplňující omezení a poznámky.

V poznámkách se vyskytují jak popisy atributů, které by mohly být nejasné, tak příklad formátu vkládaných dat.

Tabulka 3: entita zamestnanec_pozice - ciselnik

Integritní omezení	Název položky	Datový typ	Doplňující podmínky	Poznámky
PK	id_pozice	int	not null	
	nazev	varchar(20)	not null	název pozice
	poznamky	varchar(200)	Sparse null	stručný popis pozice

Zdroj: vlastní tvorba

Tabulka 4: entita psc_mesto - ciselnik

Integritní omezení	Název položky	Datový typ	Doplňující podmínky	Poznámky
PK	psc	varchar(5)	not null	např.: 67602 <-bez mezer
	mesto	varchar(25)	not null	

Zdroj: vlastní tvorba

Tabulka 5: entita stav_platby - ciselnik

Integritní omezení	Název položky	Datový typ	Doplňující podmínky	Poznámky
PK	id_stav_platby	tinyint	identity(1,1)	
	nazev	varchar(11)	not null	zaplaceno x nezaplaceno

Zdroj: vlastní tvorba

Tabulka 6: entita zpusob_doruceni - ciselnik

Integritní omezení	Název položky	Datový typ	Doplňující podmínky	Poznámky
PK	id_zpusob_doruceni	tinyint	identity(1,1)	
	nazev_popis	varchar(25)	not null	způsob doručení
	poplatek	numeric(5,2)	not null	poplatek (cena)

Zdroj: vlastní tvorba

Tabulka 7: entita zboží

Integritní omezení	Název položky	Datový typ	Doplňující podmínky	Poznámky
PK	id_zbozi	int	identity(1,1)	
	nazev	varchar(30)	not null	název (jméno)
	popis	varchar(50)	Sparse null	popis (charakteristika)
	cena_jednotkova_nakup	numeric(7,2)	not null	nákupní cena
	cena_jednotkova_prodej	numeric(7,2)	not null	prodejní cena

Zdroj: vlastní tvorba

Tabulka 8: entita typ_platby - císelník

Integritní omezení	Název položky	Datový typ	Doplňující podmínky	Poznámky
PK	id_tpu_platby	tinyint	identity(1,1)	
	nazev	varchar(8)	not null	jakým způsobem bude objednávka uhrazena

Zdroj: vlastní tvorba

Entita typ_platby může být například naplněna těmito možnostmi: hotovost, platba převodem, platební karta, dobírka, jiné.

Tabulka 9: entita zakaznik

Integritní omezení	Název položky	Datový typ	Doplňující podmínky	Poznámky
PK	id_zakaznik	int	identity(1,1)	
	nazev	varchar(25)	not null	jméno zákazníka nebo název společnosti
	kontakt_osoba_jmeno	varchar(15)		pokud stejné jako "nazev", tak neuvádíme
	kontakt_osoba_prijmeni	varchar(25)		Viz. "kontakt_osoba_jmeno"
	ulice	varchar(25)	not null	
	cp	varchar(8)	not null	např.: 1475/48b
FK(psc_mesto)	psc	varchar(5)	not null	např.: 67602 <-bez mezer
	telefon	varchar(12)	not null	např.: 420721912674
	ico	varchar(8)		
	email	varchar(60)	Sparse null	
	datum_registrace	smalldatetime	not null	2012-03-25 17:25:31

Zdroj: vlastní tvorba

Tabulka 10: entita pobočka

Integritní omezení	Název položky	Datový typ	Doplňující podmínky	Poznámky
PK	id_pobočka	int	identity(1,1)	
	nazev	varchar(25)	not null	
	ulice	varchar(25)	not null	
	cp	varchar(8)	not null	např.: 1475/48b
FK(psc_mesto)	psc	varchar(5)	not null	
	email	varchar(60)	not null	
	plan_prodeje	numeric(10,2)	not null	plány objemu tržeb - statistiky
	datum_otevreni	date	not null	datum otevření pobočky

Zdroj: vlastní tvorba

Tabulka 11: entita pobočka_telefony

Integritní omezení	Název položky	Datový typ	Doplňující podmínky	Poznámky
FK(pobočka)	id_pobočka	int	not null	
	telefon	varchar(12)	not null	Např.: 420721912674
	popis	varchar(20)	not null	Např.: servis,obchod...

Zdroj: vlastní tvorba

Tabulka 12: entita zamestnanec

Integritní omezení	Název položky	Datový typ	Doplňující podmínky	Poznámky
PK	id_zam	int	identity(1,1)	
	jmeno	varchar(15)	not null	jméno zaměstnance
	prijmeni	varchar(25)	not null	příjmení
	rodne_cislo	varchar(11)	not null	Např.: 900212/4418
FK(psc_mesto)	psc	varchar(5)	not null	např.: 67602 <-bez mezer
	ulice	varchar(25)	not null	
	cp	varchar(8)	not null	např.: 1475/48b
	telefon	varchar(12)	not null	např.: 420721927674
	country	varchar(30)	not null	Česká republika
	pracuje_od	date	not null	2012-03-25
FK(pobočka)	id_pobočka	int	not null	
FK(zamestnanec_pozice)	id_pozice	int	not null	
	poznamky	varchar(200)	Sparse null	Poznámky

Zdroj: vlastní tvorba

Tabulka 13: entita objednavky_prodej

Integritní omezení	Název položky	Datový typ	Doplňující podmínky	Poznámky
PK	id_objednavka_p rodej	int	identity(1,1)	
FK(zakaznik)	id_zakaznik	int	not null	
FK(zamestnanec)	id_zam	int	not null	
	datum_KDY	smalldatetime	not null	datum vytvoření objednávky (2012-03-25 17:25:31)
	datum_ZMENA	smalldatetime		datum editace objednávky (2012-03-25 17:42:44)
	cena	numeric(9,2)	not null	např.: 1000,00
	splatnost	date	not null	2012-04-15
FK(zpusob_doruceni)	id_zpusob_doruceni	tinyint	not null	
FK(typ_platby)	id_typu_platby	tinyint		
FK(stav_platby)	id_stav_platby	tinyint	not null default =2=NEZAPLA CENO	

Zdroj: vlastní tvorba

Tabulka 14: entita stav_expedice - ciselnik

Integritní omezení	Název položky	Datový typ	Doplňující podmínky	Poznámky
PK	id_stav_expedice	tinyint	identity(1,1)	
	nazev	varchar(12)	not null	EXPEDOVANOxNEEXPEDOVANO

Zdroj: vlastní tvorba

Tabulka 15: entita objednane_zbozi_prodej

Integritní omezení	Název položky	Datový typ	Doplňující podmínky	Poznámky
FK(objednavky_prodej)	id_objednavka_p rodej	int	not null	
FK(zbozi)	id_zbozi	int	not null	
	sleva	tinyint	not null default= 0	%
FK(stav_expedice)	id_stav_expedice	tinyint	not null default=2=NE	ANOxNE
	mnozstvi	tinyint	not null	ks

Zdroj: vlastní tvorba

Tabulka 16: entita dodavatel

Integritní omezení	Název položky	Datový typ	Doplňující podmínky	Poznámky
PK	id_dodavatel	int	identity(1,1)	
	nazev	varchar(25)	not null	název dodavatele
	kontakt_osoba_jmeno	varchar(15)		
	kontakt_osoba_prijmeni	varchar(25)		
	ulice	varchar(25)	not null	
	cp	varchar(8)	not null	např.: 1475/48b
FK(psc_mesto)	psc	varchar(5)	not null	např.: 67602
	telefon	varchar(12)	not null	např.: 420721937674
	ico	varchar(8)		
	email	varchar(60)	not null	xxxxx@xxxxx.xxx
	datum_registrace	smalldatetime	not null	datum registrace dodavatele do systemu

Zdroj: vlastní tvorba

Tabulka 17: entita objednavky_nakup

Integritní omezení	Název položky	Datový typ	Doplňující podmínky	Poznámky
PK	id_objednavka_nakup	int	identity(1,1)	
FK(dodavatel)	id_dodavatel	int	not null	
FK(zamestnanec)	id_zam	int	not null	
	datum_KDY	smalldatetime	not null	datum vytvoření objednávky (2012-03-25 17:25:31)
	cena	numeric(9,2)	not null	např.: 1000,00
	splatnost	date	not null	2012-04-15
FK(zpusob_doruceni)	id_zpusob_doruceni	tinyint	not null	
FK(typ_platby)	id_typu_platby	tinyint		
FK(stav_platby)	id_stav_platby	tinyint	not null default =2=NEZAPLACENO	ZAPLACENOxNEZAPLACENO

Zdroj: vlastní tvorba

Tabulka 18: entita objedname_zbozi_nakup

Integritní omezení	Název položky	Datový typ	Doplňující podmínky	Poznámky
FK(objednavky_nakup)	id_objednavka_nakup	int	not null	
FK(zbozi)	id_zbozi	int	not null	
	sleva	tinyint	default= 0	%
	mnozstvi	tinyint	not null	ks

Zdroj: vlastní tvorba

Tabulka 19: entita zasoby_pobocky

Integritní omezení	Název položky	Datový typ	Doplňující podmínky	Poznámky
PK	id_zaznam_zbozi_pobocka	int	identity(1,1)	
FK(zbozi)	id_zbozi	int	not null	
FK(pobocka)	id_pobocka	int	not null	
	zasoba	int	not null	ks
	min_zasoba	int		ks

Zdroj: vlastní tvorba

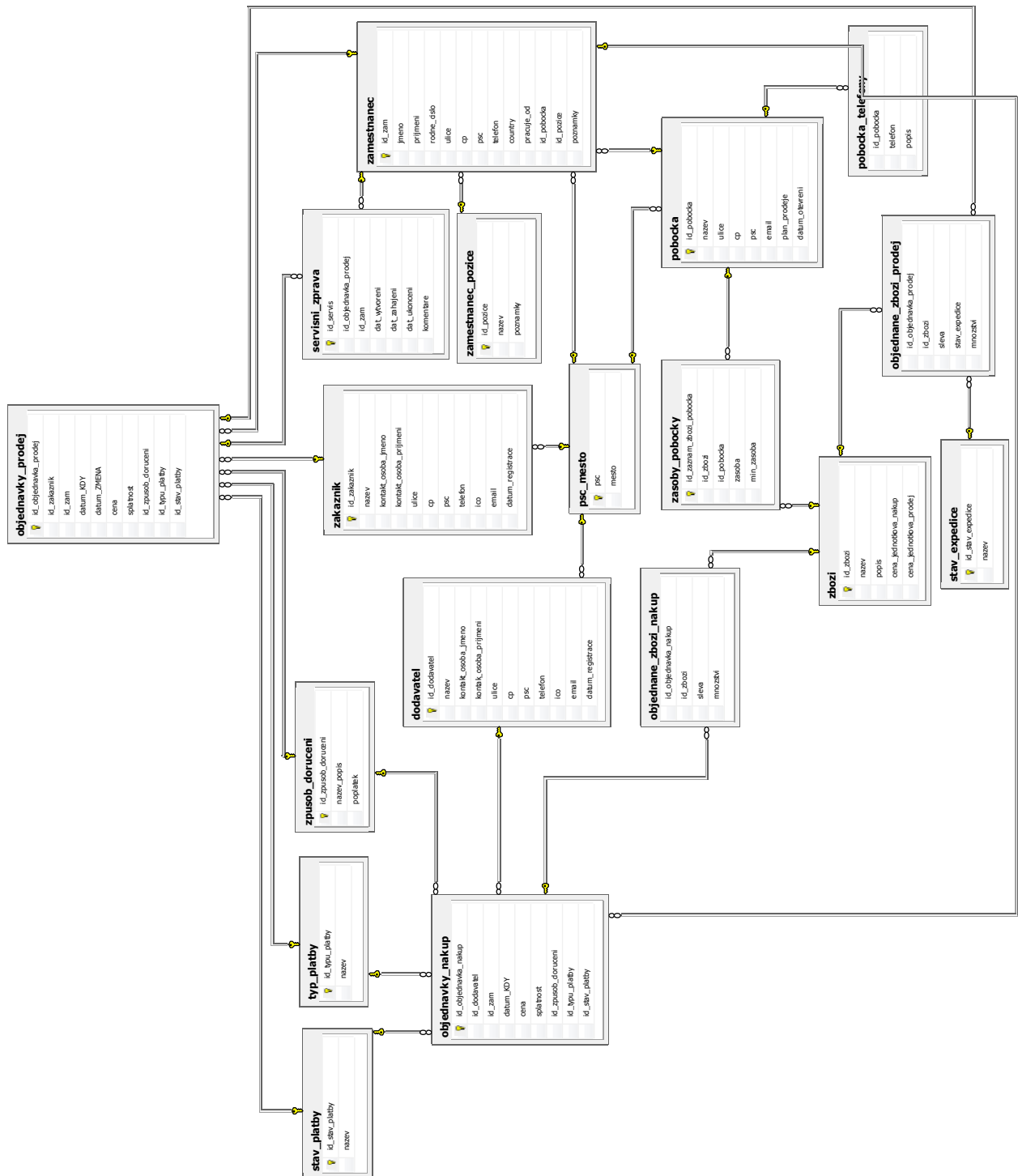
Tabulka 20: entita servisni_zprava

Integritní omezení	Název položky	Datový typ	Doplňující podmínky	Poznámky
PK	id_servis	int	identity(1,1)	
FK(objednavky_prodej)	id_objednavka_prodej	int		v případě, že je potřeba vyměnit nějaký díl,součástku..
FK(zamestnanec)	id_zam	int	not null	zaměstnanec, který servis(opravu) vyřizoval/prováděl
	dat_vytvoreni	smalldatetime	not null	datum registrace opravy (2012-03-25 17:25:31)
	dat_zahajeni	smalldatetime		datum zahájení opravy (2012-03-26 09:25:31)
	dat_ukonceni	smalldatetime		datum ukončení opravy (2012-04-03 17:25:31)
	komentare	varchar(1800)	not null	MAX JEDNA NORMOSTRANA slovní vyjádření, popis závady, způsob opravy...

Zdroj: vlastní tvorba

3.5.7 Finální E-R Diagram

Tento finální E-R diagram znázorňuje přehled propojení jednotlivých tabulek (entit) a vazby mezi nimi vyplývající z logického návrhu databáze.



Obr. 20: Finální E-R diagram

Zdroj: Vlastní tvorba

3.6 Fyzický návrh databáze

Tato část je zaměřena na vytvoření kódu databáze na základě logického návrhu. Tento kód se nachází v příloze č.1 a zde je uvedeno několik ukázek výstupu z této databáze.

3.6.1 Dotazy na výstup z vlastního řešení

V této podkapitole se zaměřuji na několik základních výstupů z databáze pomocí procedur. Dále zde demonstuji využití transakce, triggeru a pohledu (view) na konkrétních příkladech.

Kdo prováděl servis - procedura

V případě, že bude zákazník nespokojen s provedenou opravou nebo bude mít nějaký rozpor s tím proč například nebyla reklamační uznána, může kontaktovat vedoucího pobočky a po zadání čísla servisní zprávy do systému, jednoduše zjistíme kdo servis prováděl.

Po vykonání této procedury není vidět komentář k servisní zprávě. Není považován v prvotní chvíli za důležitý, lze si však zobrazit samostatnou zprávu, kde již bude komentář k nahlédnutí.

--Vytvoření procedury

go

create procedure servis_kdo (@cislo_servis int)

as

begin

select s.id_servis as 'ID servisni zprávy', s.id_objednavka_prodej as 'ID prodejní objednávky', s.dat_zahajeni as 'zahájení opravy/reklamačního řízení', s.dat_ukonceni as 'datum ukončení', p.id_zam as 'ID zaměstnance', p.jmeno as 'Jméno', p.prijmeni as 'Příjmení'

from servisni_zprava s, zamestnanec p

where @cislo_servis = s.id_servis and s.id_zam=p.id_zam

end

```

go
--Zavolání procedury
exec servis_kdo '1'
--Odvolání/smazání
drop procedure servis_kdo

```

Rychlé vyhledání kontaktu na dodavatele - procedura

Tato procedura (funkce databáze) umožňuje po zadání názvu dodavatele nebo jména konkrétní kontaktní osoby v dodavatelské firmě vyhledat telefon a email, aby společnost byla schopna v co nejkratším čase zjistit podrobnosti(dodací lhůta, cena, ...) o daném zboží.

```

--Vytvoření procedury
go
create procedure najdi_kontakt_dodavatele (@nazev varchar(25),@jmeno
varchar(15),@prijmeni varchar(25))
as
select id_dodavatel as 'ID dodavatele', nazev as 'Název', kontakt_osoba_jmeno as
'Jméno kontaktní osoby', kontak_osoba_prijmeni as 'Příjmení kontaktní osoby', telefon,
email
from dodavatel
where @nazev=nazev or (@jmeno = kontakt_osoba_jmeno and @prijmeni =
kontak_osoba_prijmeni)
or (@nazev=nazev and @jmeno = kontakt_osoba_jmeno and @prijmeni =
kontak_osoba_prijmeni)
go

--Zavolání procedury
exec najdi_kontakt_dodavatele '100Mega distribution',null,null

--Odvolání/smazání procedury
drop procedure najdi_kontakt_dodavatele

```

Nastavení prodejní ceny u všeho zboží (7% marže) - transakce

V začátcích prodeje, nebo při cenových úpravách lze pomocí této transakce navýšit cenu zboží o určité procento oproti nákupní ceně. Po malé úpravě lze upravovat ceny u jednotlivých zboží.

```
BEGIN TRAN uprava_prodejni_ceny
UPDATE zbozi SET cena_jednotkova_prodej=1.07*cena_jednotkova_nakup
SELECT * FROM zbozi
--Odvolání transakce
ROLLBACK TRAN uprava_prodejni_ceny
SELECT * FROM zbozi
```

Trigger - výpis hlášení, že byla nastavena/upravena hodnota cen(y) v tabulce zboží

Po provedení transakce, nastavení nebo uprav cen se spustí tento trigger (hláška).

```
go
create trigger editace ON Zbozi
after update
as
begin
    PRINT 'cena(ceny) byly upravena(upraveny)'
end
go
--Zrušení triggeru
alter table Zbozi DISABLE TRIGGER editace
```

Trigger - výpis hlášení, že byl vložen zákazník

Po vložení nového zákazníka do tabulky se zobrazí hláška "nový zákazník byl vložen do databáze".

```
go
create trigger Potvrzeni ON Zakaznik
after insert
as
begin
    PRINT 'Nový zákazník byl vložen do databáze'
end
go
--Zrušení triggeru
alter table Zakaznik DISABLE TRIGGER Potvrzeni
```

Zboží skladem na pobočce Moravské Budějovice - pohled (view)

Tento pohled nám slouží k zobrazení skladových položek na pobočce v Moravských Budějovicích. Kód je vytvořen pouze pro tuto konkrétní pobočku, nicméně po drobné modifikaci jej lze použít i pro jiné pobočky.

```
--Vytvoření pohledu
go
create view skladove_polozky as
SELECT      zbozi.id_zbozi AS 'ID zboží', zbozi.nazev AS 'Název', zbozi.popis AS
'Popis', zbozi.cena_jednotkova_prodej AS 'Prodejní cena [ks]',
            zasoby_pobocky.zasoba AS 'Skladem', zasoby_pobocky.min_zasoba AS
'Minimální zásoba', psc_mesto.mesto AS 'Město'
FROM        zasoby_pobocky INNER JOIN
            zbozi ON zasoby_pobocky.id_zbozi = zbozi.id_zbozi INNER JOIN
            pobocka ON zasoby_pobocky.id_pobocka = pobocka.id_pobocka
INNER JOIN
            psc_mesto ON pobocka.psc = psc_mesto.psc
```

```
WHERE      (psc_mesto.mesto = 'Moravské Budějovice')  
go
```

--Zobrazení dat v pohledu

```
select * from skladove_polozky
```

--Odstranění pohledu

```
drop view skladove_polozky
```


3.7 Přínos návrhu řešení

Cílem této bakalářské práce bylo vytvořit SQL databázi pro podporu činností v malé IT firmě. Vzhledem k tomu, že firma dosud nevyužívá žádný informační systém, který by tuto oblast pokrýval, je přínos mnou navrhovaného řešení velkým krokem kupředu. Samozřejmě, že se jedná pouze o návrh databáze, kterého se musí ještě chopit zkušený programátor, aby bylo možno tento systém aplikovat v praxi. Nicméně si myslím, že jsem poskytl slušný základ pro systém, který by měl společnosti plně vyhovovat.

Navrhovaná databáze má potenciál zbavit firmu problémů, které doposud při práci s daty a informacemi měla. Díky tomuto řešení se společnost zbaví mnoha problému, jak v komunikaci mezi jednotlivými odděleními, tak neefektivním zpracování, nekonzistentnosti dat a také problémů s archivací nejednotných záznamů.

Mnou navrhované řešení SQL databáze poskytne ucelený a jednotný pohled na důležité provozní informace. Platforma MS SQL server 2008, na které je databáze navrhuta a na které bude fungovat, umožňuje nastavení přístupových práv, mnoho možností archivace a vytváření pravidelných záloh. Samotný návrh databáze neumožňuje uživateli uložit záznamy bez řádného vyplnění povinných atributů. Dalším přínosem je značné navýšení efektivity, kdy se urychlí všechny probíhající činnosti. Nyní bude možno všechny objednávky vyřizovat pohodlně v reálném čase přes informační systém, stejně jako reklamace a další činnosti, které firma provádí za účelem dosažení zisku. Další výhodou bude dokonalý přehled o skladech. Firma ihned bude vědět jaké zboží, množství a ve kterém skladě (pobočce) se nachází.

Nakonec je nutno podotknout, že takto navrhovaný systém je možno propojit s již zakoupeným mzdovým a účetním modulem systému S3 money a vytvořit, tak jednotný ucelený systém, který ve společnosti doposud chyběl.

Investice do tohoto nového systému je vzhledem k oblasti působení společnosti minimální. Přesto se však tato investice navrátí již v době implementace a to zejména ve zkrácení doby vykonávání všech činností a následné administrativy což přispěje k vyšší efektivitě práce a množnosti pokrýt požadavky více zákazníků v kratším čase.

Závěr

Cílem této práce bylo vytvořit SQL databázi na podporu činností v malé IT firmě zabývající se obchodem, poskytováním služeb a poradenství v oblasti IT. Mezi dílčí cíle bych zahrnul zejména analýzu klíčových činností společnosti a splnění požadavků, které byly kladeny na vytvářenou databázovou strukturu. Mezi tyto požadavky patří přehlednost, aktuálnost a zabezpečení uchovávaných dat.

Práce je rozdělena do tří stěžejních kapitol. První z nich poskytuje teoretické pozadí této práce, které bylo nezbytné k vytvoření výsledné databázové struktury.

Nezbytná kapitola pro samotný návrh řešení je bezpochyby i kapitola druhá, kde jsem zanalyzoval současný stav ve společnosti. V této analýze jsem okrajově rozebral vnitřní a vnější ekonomické prostředí společnosti a také současný stav práce s informacemi a daty.

Třetí kapitola obsahuje vlastní návrh struktury databáze, který stavěl na požadavcích zaměstnanců a majitele společnosti. Podstatnou část zde hrála analýza činností společnosti, které vzhledem ke stáří společnosti a neexistenci informačního systému neměli jednotnou ucelenou strukturu, které se ovšem podařilo dosáhnout a popsat je vývojovými diagramy. Dále proběhla analýza toků informací ve firmě, kde jsme se zaměřili zejména na klíčové firemní činnosti. Tento tok informací je znázorněn pomocí DFD diagramu.

Po té následovalo datové modelování, kde došlo k identifikaci základních entit, vazeb mezi nimi z čehož jsem vytvořil základní E-R diagram. Ze základního E-R diagramu bylo jasné, že musí proběhnout dekompozice některých vazeb. Po této dekompozici jsem mohl definovat jednotlivé entity databáze a vygenerovat finální E-R diagram. Všechny tyto činnosti byly nezbytné pro vytvoření databáze pomocí jazyka SQL.

Výsledkem mé práce je databázová struktura, která bude tvořit jádro nového, ve společnosti prvního, informačního systému pokrývajícího činnosti v ní probíhající. Pokud se tento mnou položený základní kámen dostane do interakce se zkušeným programátorem, je zde velká šance vytvořit systém, který by bylo možno s drobnými modifikacemi nasazovat ve společnostech stejného rozsahu a obchodního zaměření jako je společnost mnou analyzovaná.

Literatura

Knižní zdroje

- [1] CONOLLY, Thomas; BEGG, Carolyn; HOLOWCZAK, Richard. *Mistrovství - Databáze : Profesionální průvodce tvorbou efektivních databází*. Brno : Computer Press, 2009. 584 s. ISBN 978-80-251-2328-7.
- [2] DOSTÁL, J. *Hardware moderního počítače*. Olomouc: UP, 2011. 77 s. ISBN 978-80-244-2787-4.
- [3] HOTEK, M. *Microsoft SQL Server 2008: krok za krokem*. 1. vydání. Brno: Computer Press, 2009. 488 s. ISBN 978-80-251-2466-6.
- [4] KOCH, M.; NEUWIRTH, B. *Datové a funkční modelování*. 3. přepracované vydání. Brno: Akademické nakladatelství Cerm, 2008. 121s. ISBN 978-80-214-3731-9.
- [5] LACKO, L. *Jak vyžrát na SQL Server 2008*. Brno: Computer Press, 2009. 469 s. ISBN 978-80-251-2101.
- [6] MOLINARO, A. *SQL: Kuchařka programátora*. Brno: Computer Press, 2009. 576 s. ISBN 978-80-251-2617-2.
- [7] RIORDAN, Rebecca M. *Vytváření relační databázové aplikace*. Praha : Computer Press, 2000. 294 s. ISBN 80-7226-360-9.

Online zdroje

- [8] KOLÁŘ, Petr. *Operační systémy* [online]. 2005 [cit. 2012-05-12]. <http://www.nti.tul.cz/~kolar/os/>
- [9] KOSEK, Jiří. *Lehký úvod do SQL* [online]. 1999 [cit. 2012-04-27]. <http://www.kosek.cz/clanky/iweb/13.html>
- [10] RYDVAL, Slávek. *Historie jazyka SQL* [online]. 2005 [cit. 2011-11-26]. <http://www.rydval.cz/phprs/view.php?cislocclanku=2005123125>
- [11] SKŘIVAN, Jaromír. *Databáze a jazyk SQL* [online]. 2000 [cit. 2012-05-11]. <http://interval.cz/clanky/databaze-ajazyk-sql/>
- [12] W3schools.com [online]. 1999, 2010 [cit. 2011-11-24]. *SQL Tutorial*. <http://www.w3schools.com/sql/default.asp>

Seznam použitých zkratk

DFD diagram	Data Flow Diagram, Diagram toku dat
E-R diagram	Entity-Relationship diagram, Entito-Relační diagram
FK	Foreign key, Cizí klíč
IT	Information technology, informační technologie
PK	Primary key, Primární klíč
SQL	Structured Query Language, strukturovaný dotazovací jazyk

Seznam obrázků

Obr. 1: Tři základní požadavky na informaci	11
Obr. 2: Popis relace(entity)	15
Obr. 3: Logo společnosti.....	22
Obr. 4: Organizační struktura	23
Obr. 5: Použité značky vývojových diagramů.....	30
Obr. 6: Vytvoření prodejní objednávky	31
Obr. 7: Vytvoření nákupní objednávky	33
Obr. 8: Přidání nového zaměstnance	34
Obr. 9: Průběh servisem.....	35
Obr. 10: DFD diagram celého systému	36
Obr. 11: Základní E-R diagram	39
Obr. 12: Obecné schéma dekompozice.....	40
Obr. 13: Vztah entit zákazník - zboží	41
Obr. 14: Dekompozice entit zákazník - zboží.....	41
Obr. 15: Vztah entit zaměstnanec - zboží	42
Obr. 16: Dekompozice entit zaměstnanec - zboží	42
Obr. 17: Vztah entit pobočka - zboží	42
Obr. 18: Dekompozice entit pobočka - zboží	43
Obr. 19: Schéma konečné dekompozice	43
Obr. 20: Finální E-R diagram	50

Seznam tabulek

Tabulka 1: definice základních entit	37
Tabulka 2: identifikace relací mezi základními entitami	38
Tabulka 3: entita zamestnanec_pozice - ciselnik	44
Tabulka 4: entita psc_mesto - ciselnik	44
Tabulka 5: entita stav_platby - ciselnik	44
Tabulka 6: entita zpusob_doruceni - ciselnik	44
Tabulka 7: entita entita zbozi	45
Tabulka 8: entita typ_platby - ciselnik	45
Tabulka 9: entita zakaznik	45
Tabulka 10: entita pobočka	46
Tabulka 11: entita pobočka_telefony	46
Tabulka 12: entita zamestnanec	46
Tabulka 13: entita objednavky_prodej	47
Tabulka 14: entita stav_expedice - ciselnik	47
Tabulka 15: entita objednane_zbozi_prodej	47
Tabulka 16: entita dodavatel	48
Tabulka 17: entita objednavky_nakup	48
Tabulka 18: entita objednane_zbozi_nakup	49
Tabulka 19: entita zasoby_pobocky	49
Tabulka 20: entita servisni_zprava	49

Přílohy

Příloha č.1 : MS SQL query – zdrojový kód fyzického návrhu databáze

```
create database FIRMA
```

```
go
```

```
use FIRMA
```

```
go
```

```
-- Císelník zaměstnanci-pozice
```

```
CREATE TABLE zamestnanec_pozice
```

```
(
```

```
    id_pozice INT identity(1,1) NOT NULL,
```

```
    nazev VARCHAR(20) NOT NULL,
```

```
    poznamky VARCHAR(200) SPARSE NULL,
```

```
    PRIMARY KEY (id_pozice)
```

```
);
```

```
CREATE UNIQUE INDEX Izamestnanec_pozice ON zamestnanec_pozice
```

```
(id_pozice);
```

```
go
```

```
-- Tabulka mesto-seznam psc -> mesto
```

```
CREATE TABLE psc_mesto
```

```
(
```

```
    psc VARCHAR(5) NOT NULL,
```

```
    mesto VARCHAR(25) NOT NULL,
```

```
    PRIMARY KEY (psc)
```

```
);
```

```
go
```

```
CREATE UNIQUE INDEX Ipsc_mesto ON psc_mesto (psc);
```

```

-- Tabulka stav_platby
CREATE TABLE stav_platby
(
    id_stav_platby TINYINT identity(1,1) NOT NULL,
    nazev VARCHAR(11) NOT NULL,
    PRIMARY KEY (id_stav_platby)
);

CREATE UNIQUE INDEX Istav_platby ON stav_platby (id_stav_platby);
go

-- Ciselnik zpusob_doruceni
CREATE TABLE zpusob_doruceni
(
    id_zpusob_doruceni TINYINT identity(1,1) NOT NULL,
    nazev_popis VARCHAR(25) NOT NULL,
    poplatek numeric(5,2) NOT NULL,
    PRIMARY KEY (id_zpusob_doruceni)
);

CREATE UNIQUE INDEX Izpusob_doruceni ON zpusob_doruceni
(id_zpusob_doruceni);
go

-- Tabulka zbozi
CREATE TABLE zbozi
(
    id_zbozi INT identity(1,1) NOT NULL,
    nazev VARCHAR(30) NOT NULL,
    popis VARCHAR(50) SPARSE NULL,
    cena_jednotkova_nakup NUMERIC(7,2) NOT NULL,
    cena_jednotkova_prodej NUMERIC(7,2) NOT NULL,

```



```

PRIMARY KEY (id_zbozi)
);

CREATE UNIQUE INDEX Izbozi ON zbozi (id_zbozi);
go

-- Císelník typ_platby
CREATE TABLE typ_platby
(
    id_typu_platby TINYINT identity(1,1) NOT NULL,
    nazev VARCHAR(15) NOT NULL,
    PRIMARY KEY (id_typu_platby)
);

CREATE UNIQUE INDEX Ityp_platby ON typ_platby (id_typu_platby);
go

-- Tabulka zakaznik
CREATE TABLE zakaznik
(
    id_zakaznik INT identity(1,1) NOT NULL,
    nazev VARCHAR(25) NOT NULL,
    kontakt_osoba_jmeno VARCHAR(15),
    kontakt_osoba_prijmeni VARCHAR(25),
    ulice VARCHAR(25),
    cp VARCHAR(8),
    psc VARCHAR(5) NOT NULL,
    telefon VARCHAR(12) NOT NULL,
    ico VARCHAR(8),
    email VARCHAR(60) SPARSE NULL,
    datum_registrace SMALLDATETIME NOT NULL,
    PRIMARY KEY (id_zakaznik),

```

```
FOREIGN KEY (psc) REFERENCES psc_mesto ON DELETE CASCADE  
);
```

```
CREATE UNIQUE INDEX Izakaznik ON zakaznik (id_zakaznik);  
go
```

```
-- Tabulka pobočka
```

```
CREATE TABLE pobočka
```

```
(  
    id_pobočka INT identity(1,1) NOT NULL,  
    nazev VARCHAR(25) NOT NULL,  
    ulice VARCHAR(25) NOT NULL,  
    cp VARCHAR(8) NOT NULL,  
    psc VARCHAR(5) NOT NULL,  
    email VARCHAR(60) NOT NULL,  
    plan_prodeje NUMERIC(10,2),  
    datum_otevreni DATE NOT NULL,  
    PRIMARY KEY (id_pobočka),  
    FOREIGN KEY (psc) REFERENCES psc_mesto ON DELETE CASCADE  
);
```

```
CREATE UNIQUE INDEX Ipobočka ON pobočka (id_pobočka);  
go
```

```
-- Tabulka pobočka_telefony
```

```
CREATE TABLE pobočka_telefony
```

```
(  
    id_pobočka INT NOT NULL,  
    telefon VARCHAR(12) NOT NULL,  
    popis VARCHAR(20) NOT NULL,  
    FOREIGN KEY (id_pobočka) REFERENCES pobočka ON DELETE CASCADE  
);
```

go

-- Tabulka zamestnanec

CREATE TABLE zamestnanec

```
(
  id_zam INT identity(1,1) NOT NULL,
  jmeno VARCHAR(15) NOT NULL,
  prijmeni VARCHAR(25) NOT NULL,
  rodne_cislo varchar(11) NOT NULL,
  ulice VARCHAR(25) NOT NULL,
  cp VARCHAR(8) NOT NULL,
  psc varchar(5) NOT NULL,
  telefon VARCHAR(12) NOT NULL,
  country VARCHAR(30) NOT NULL,
  pracuje_od DATE NOT NULL,
  id_pobocka INT NOT NULL,
  id_pozice INT NOT NULL,
  poznamky VARCHAR(200) SPARSE NULL,
  PRIMARY KEY (id_zam),
  FOREIGN KEY (psc) REFERENCES psc_mesto,
  FOREIGN KEY (id_pobocka) REFERENCES pobocka,
  FOREIGN KEY (id_pozice) REFERENCES zamestnanec_pozice
);
```

CREATE UNIQUE INDEX Izamestnanec ON zamestnanec (id_zam);

go

-- Tabulka objednavky_prodej

CREATE TABLE objednavky_prodej

```
(
  id_objednavka_prodej INT identity(1,1) NOT NULL,
  id_zakaznik INT NOT NULL,
```

```

id_zam INT NOT NULL,
datum_KDY SMALLDATETIME NOT NULL,
datum_ZMENA SMALLDATETIME,
cena NUMERIC(9,2) NOT NULL,
splatnost DATE NOT NULL,
id_zpusob_doruceni TINYINT NOT NULL,
id_typu_platby TINYINT,
id_stav_platby TINYINT NOT NULL DEFAULT '2',
PRIMARY KEY (id_objednavka_prodej),
FOREIGN KEY (id_zakaznik) REFERENCES zakaznik,
FOREIGN KEY (id_zam) REFERENCES zamestnanec,
FOREIGN KEY (id_zpusob_doruceni) REFERENCES zpusob_doruceni,
FOREIGN KEY (id_typu_platby) REFERENCES typ_platby,
FOREIGN KEY (id_stav_platby) REFERENCES stav_platby
);

```

```

CREATE UNIQUE INDEX Iobjednavky_prodej ON objednavky_prodej
(id_objednavka_prodej);
go

```

```

-- Ciselnik stav_expedice
CREATE TABLE stav_expedice
(
    id_stav_expedice TINYINT identity(1,1) NOT NULL,
    nazev VARCHAR(12) NOT NULL,
    PRIMARY KEY (id_stav_expedice)
);

```

```

CREATE UNIQUE INDEX Istav_expedice ON stav_expedice (id_stav_expedice);
go

```

```

-- Tabulka objednane_zbozi_prodej

```

```

CREATE TABLE objednane_zbozi_prodej
(
    id_objednavka_prodej INT NOT NULL,
    id_zbozi INT NOT NULL,
    sleva TINYINT NOT NULL, -- DEFAULT '2',
    stav_expedice TINYINT NOT NULL, -- DEFAULT '2',
    mnozstvi TINYINT NOT NULL,
    FOREIGN KEY (id_objednavka_prodej) REFERENCES objednavky_prodej ON
DELETE CASCADE,
    FOREIGN KEY (id_zbozi) REFERENCES zboží ON DELETE CASCADE,
    FOREIGN KEY (stav_expedice) REFERENCES stav_expedice
);
go

```

-- Tabulka dodavatel

```

CREATE TABLE dodavatel
(
    id_dodavatel INT identity(1,1) NOT NULL,
    nazev VARCHAR(25) NOT NULL,
    kontakt_osoba_jmeno VARCHAR(15),
    kontak_osoba_prijmeni VARCHAR(25),
    ulice VARCHAR(25) NOT NULL,
    cp VARCHAR(8) NOT NULL,
    psc VARCHAR(5) NOT NULL,
    telefon VARCHAR(12) NOT NULL,
    ico VARCHAR(8),
    email VARCHAR(60) NOT NULL,
    datum_registrace SMALLDATETIME,
    PRIMARY KEY (id_dodavatel),
    FOREIGN KEY (psc) REFERENCES psc_mesto ON DELETE CASCADE,
);

```

```

CREATE UNIQUE INDEX Idodavatel ON  dodavatel (id_dodavatel);

go

-- Tabulka objednavky_nakup
CREATE TABLE objednavky_nakup
(
    id_objednavka_nakup INT identity(1,1) NOT NULL,
    id_dodavatel INT NOT NULL,
    id_zam INT NOT NULL,
    datum_KDY SMALLDATETIME NOT NULL,
    cena NUMERIC(9,2) NOT NULL,
    splatnost DATE NOT NULL,
    id_zpusob_doruceni TINYINT NOT NULL,
    id_typu_platby TINYINT,
    id_stav_platby TINYINT NOT NULL DEFAULT '2',
    PRIMARY KEY (id_objednavka_nakup),
    FOREIGN KEY (id_dodavatel) REFERENCES dodavatel,
    FOREIGN KEY (id_zam) REFERENCES zamestnanec,
    FOREIGN KEY (id_zpusob_doruceni) REFERENCES zpusob_doruceni,
    FOREIGN KEY (id_typu_platby) REFERENCES typ_platby,
    FOREIGN KEY (id_stav_platby) REFERENCES stav_platby
);

CREATE UNIQUE INDEX Iobjednavky_nakup ON  objednavky_nakup
(id_objednavka_nakup);

go

-- Tabulka objednane_zbozi_nakup
CREATE TABLE objednane_zbozi_nakup
(
    id_objednavka_nakup INT NOT NULL,
    id_zbozi INT NOT NULL,

```

```

sleva TINYINT NOT NULL DEFAULT '0',
mnozstvi TINYINT NOT NULL,
FOREIGN KEY (id_objednavka_nakup) REFERENCES objednavky_nakup ON
DELETE CASCADE,
FOREIGN KEY (id_zbozi) REFERENCES zboží ON DELETE CASCADE,
);
go

```

```

-- Tabulka zasoby_pobocky
CREATE TABLE zasoby_pobocky
(
id_zaznam_zbozi_pobocka INT identity(1,1) NOT NULL,
id_zbozi INT NOT NULL,
id_pobocka INT NOT NULL,
zasoba INT NOT NULL,
min_zasoba INT,
PRIMARY KEY (id_zaznam_zbozi_pobocka),
FOREIGN KEY (id_zbozi) REFERENCES zboží,
FOREIGN KEY (id_pobocka) REFERENCES pobocka
);

```

```

CREATE UNIQUE INDEX Izasoby_pobocky ON zasoby_pobocky
(id_zaznam_zbozi_pobocka);
go

```

```

-- Tabulka servisni_zprava
CREATE TABLE servisni_zprava
(
id_servis INT identity(1,1) NOT NULL,
id_objednavka_prodej INT,
id_zam INT NOT NULL,
dat_vytvoreni SMALLDATETIME NOT NULL,

```

```
dat_zahajeni SMALLDATETIME,  
dat_ukoncení SMALLDATETIME,  
komentare VARCHAR(1800) NOT NULL,  
PRIMARY KEY (id_servis),  
FOREIGN KEY (id_objednavka_prodej) REFERENCES objednavky_prodej,  
FOREIGN KEY (id_zam) REFERENCES zamestnanec  
);  
  
CREATE UNIQUE INDEX Iservisni_zprava ON servisni_zprava (id_servis);  
go
```